APPLICATIONS OF DATA AND INFORMATION FUSION

FOO PEK HUI

NATIONAL UNIVERSITY OF SINGAPORE
2008

APPLICATIONS OF DATA AND INFORMATION FUSION

FOO PEK HUI

(M.Sc., B.Sc.(Hons.), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF PHYSICS

NATIONAL UNIVERSITY OF SINGAPORE

2008

Acknowledgements

The author would like to express sincere gratitude to

- the thesis advisor, Dr Ng Gee Wah, for his patient guidance and tolerance throughout this candidature;
- colleagues cum mentors at DSO National Laboratories, for their helpful discussions and advice;
- the thesis examiners, for their constructive comments and suggestions on improving this thesis;
- administrative and technical staff from the National University of Singapore, for their assistance on various matters;
- everyone else who provided motivation for the completion of this research.

This research was partially financed by the National University of Singapore and DSO National Laboratories.

Contents

Summary

Data and information fusion is a multidisciplinary field of research that is gaining increasing importance. This is engendered by voluminous data and information flow in various application areas from both the military and civilian sectors, as well as ubiquity and advances in communication, computing and sensor technology. In this project, we investigate various issues and applications of data and information fusion.

Firstly, we review several existing models for data and information fusion. Research focus is currently shifting from low-level information fusion, an increasingly mature area, towards the less developed area of high-level information fusion. We do an extensive survey of the existing literature on high-level information fusion, indicate/compare some of the existing approaches and discuss some relevant application areas.

Secondly, we consider the topic of target tracking. We derive an algorithm for state estimation via the combination of existing filtering techniques. The proposed approach is an Interacting Multiple Model (IMM) algorithm that makes use of various combinations of extended Kalman filters, unscented Kalman filters and particle filters for the models. Two manœuvring target tracking problems are considered. In the first problem, the IMM algorithm variants are implemented for tracking target motion in three-dimensional space. In the second problem, extended Kalman filters, unscented Kalman filters and the IMM variants are applied to the localization and tracking of a target in a horizontal plane, using a Time Difference of Arrival system. Experimental test results provide indications that it is possible to attain superior performance in state estimation with IMM algorithm variants that require relatively moderate computational load/costs. We also compare the performance of the nonlinear filters and IMM algorithms on a real-world problem on pricing financial options.

Thirdly, we describe an approach for intent inference based on the analysis of flight profiles. The proposed method, which utilizes IMM-based state estimation and fuzzy inference mechanism, is applied to two problems. The first task is to determine the possibility of weapon delivery by an attack aircraft under military surveillance. The second is to determine the possibility of non-conformance in the behaviour of an aircraft being monitored by an air traffic control system. Simulation test results show that our approach provides timely inference and demonstrates practicability as a useful aid for human cognition and critical decision making. Next, we consider using alternative IMM algorithm variants for state estimation in the proposed intent inference method. Numerical test results are compared to identify IMM variants which perform well in state estimation, subject to constraints on computation time required for reaction.

List of Tables

List of Figures

List of Symbols

Symbol	**Definition**
C	Total number of independent Monte Carlo simulation runs.
$E[\cdot]$	Expectation.
$F(k)$, $G(k)$	Jacobians of the process equation at time step k.
$H(k)$	Jacobian of the measurement equation at time step k.
$I_{m\times n}$	$m \times n$ matrix with ones on the diagonal and zeros elsewhere. When $m = n$, the matrix is an identity matrix and is written as I_n.
L	Total number of points on a target trajectory.
$M_j(k)$	Model j at time step k.
$\mathcal{N}(x; \mu, \Sigma)$	Probability density function (or density) of a multivariate Gaussian (normal) random variable x with mean μ and covariance Σ.
$\mathcal{N}(\mu, \Sigma)$	Multivariate Gaussian distribution with mean μ and covariance Σ.
$\mathcal{N}(x; \mu, \sigma^2)$	Probability density function of a Gaussian random variable x with mean μ and variance σ^2 (standard deviation σ).
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2 (standard deviation σ).
N_s	Number of samples/particles used in a particle filter.
N_{eff}	Effective sample size.
$\hat{N}_{\text{eff}}$	Estimate of effective sample size.
$O(\cdot)$	Order of magnitude of.
P_k	State error covariance associated with X_k.
$P_k^{(i)}$	State error covariance associated with $X_k^{(i)}$.
$\hat{P}_k$	State error covariance associated with $\hat{X}_k$.
$\hat{P}(k\|l)$	State error covariance associated with $\hat{x}(k\|l)$.
$\hat{P}_j(k\|k)$	State error covariance associated with $\hat{x}_j(k\|k)$.
$\hat{P}_j^0(k-1\|k-1)$	State error covariance associated with $\hat{x}_j^0(k-1\|k-1)$.
$\text{Prob}(E)$	Probability of event E.
$Q(k)$	Process noise covariance/correlation matrix at time step k.
$R(k)$	Measurement noise covariance matrix at time step k.
T	Sampling interval (time interval between successive scans) of a sensor.
$\mathcal{U}(A)$	Uniform distribution on A.

Symbol	**Definition**
$\mathrm{Var}(\cdot)$	Variance.
X_k	State vector at time step k.
$X_k^{(i)}$	The i-th sample state at time step k.
$\hat{X}_k$	State estimate at time step k.
$X_{0:k}$	State sequence through time step k.
Z_k	Measurement vector at time step k.
$Z_{1:k}$	Measurement sequence through time step k.
$\delta(\cdot)$	Dirac delta measure (or Dirac (impulse) delta function).
$\det(M)$	Determinant of a square matrix M.
e_k	Input information vector at time step k.
$f(\cdot)$	System transition function.
$g(\cdot)$	Process noise input function.
$h(\cdot)$	Measurement function.
m_k	Modal state of the system at time step k.
n_e	Dimension of the input information vector.
n_v	Dimension of the measurement noise vector.
n_w	Dimension of the process noise vector.
n_x	Dimension of the state vector.
n_z	Dimension of the measurement vector.
$p(\cdot)$	Probability density function.
$p(\cdot\|\cdot)$ or $q(\cdot\|\cdot)$	Conditional probability density function.
$q(\cdot)$	Proposal distribution (or importance sampling distribution or importance density function).
r	Number of models used in the IMM algorithm.
t_k	A continuous-time instant with time index k assigned.
$\mathrm{trace}(M)$	Sum of the diagonal elements of matrix M.
v_k	Measurement noise vector at time step k.
w_k	Process noise vector at time step k.
$w_k^{(i)}$	Importance weight corresponding to $X_k^{(i)}$.
$\hat{x}(k\|l)$	State estimate at time step k conditioned on $Z_{1:l}$.
$\hat{x}_j(k\|k)$	State estimate in $M_j(k)$.
$\hat{x}_j^0(k-1\|k-1)$	Mixed initial state estimate in $M_j(k)$.
$0_{m\times n}$	$m \times n$ matrix of zeros. When $m = n$, the matrix is written as 0_n.
$\|\cdot\|_2$	Euclidean norm.

List of Acronyms

Acronym	Definition
2D	Two-dimensional
3D	Three-dimensional
3DTR	Three-dimensional turning rate
AGL	Above ground level
APF	Auxiliary particle filter
ASIR	Auxiliary sampling importance resampling
ATC	Air traffic control
ATM	Air traffic management
BBN	Bayesian belief network
C2	Command and control
C4I	Command, control, communications, computers and intelligence
C4ISR	Command, control, communications, computers, intelligence, surveillance and reconnaissance
CA	Constant acceleration
COA	Course of action
CSW	Cumulative sum of normalized weights
CT	Coordinated turn
CV	Constant velocity
DF	Data fusion
DFIG	Data Fusion Information Group
DIF	Data and information fusion
D-S	Dempster-Shafer
EKF	Extended Kalman filter
EKPF	Extended Kalman particle filter
EL	Electrolevel
EW	Early warning
FIS	Fuzzy inference system
GMTI	Ground moving target indicator
GPF	Gaussian particle filter

Acronym	Definition
HCI	Human-computer interaction
HRR	High range resolution
IEKF	Iterated extended Kalman filter
IF	Information fusion
IID (or i.i.d.)	Independent and identically distributed
IMM	Interacting multiple model
IMU	Inertial measurement unit
INTEL	Intelligence
JDL	Joint Directors of Laboratories
KCAS	Knots calibrated airspeed
KF	Kalman filter
LSI	Location sensitivity index
MAP	Maximum a posteriori
MC	Monte Carlo
MFR	Multi function radar
MISE	Mean integrated square error
MSM	Multi-sensor management
MTI	Moving target indicator/indication
NBD	Network-based defence
NCW	Network-centric warfare
OODA	Observe, orient, decide, and act
PA	Performance assessment
PDF (or pdf)	Probability density function
PDP	Pull-down point
PE	Performance evaluation
PF	Particle filter
PM	Perception management
PUP	Pull-up point, pop-up point or pop point
RADAR	Radio detecting and ranging
RM	Resource management
RMSE	Root mean square error
RP	Release point
RPF	Regularized particle filter
RVM	Relevance vector machine
SA	Situation assessment
SAR	Synthetic aperture radar
SAW	Situation awareness
SDP	Stochastic dynamic programming

Acronym	**Definition**
SIG	Signal
SIR	Sampling importance resampling
SIS	Sequential importance sampling
SM	Sensor management
SONAR	Sound navigation and ranging
SPF	Standard particle filter
SRM	Sensor resource management
STA	Situation/threat assessment
SVM	Support vector machine
TA	Threat assessment
TDOA	Time difference of arrival
TGT	Target
TP	Track point
TRIP	Transformation of Requirements for the Information Process
UKF	Unscented Kalman filter
UPF	Unscented particle filter

Chapter 1

Introduction

Data and information fusion is a multilevel, multifaceted process of combining data and information from one or more sources to estimate or predict the states of entities in an environment over time. In general, physical states are considered. For entities in the form of information systems or sentient beings, informational states and perceptual states, as well as their relations to the physical states, may also be relevant for consideration. Informational states are data available to the target of interest. Perceptual states are a target's own estimate of the environmental state.

Data and information fusion techniques were first introduced to the research community in the 1970s. The initial applications were in the military sector [122]: ocean surveillance, air-to-air and surface-to-air defence, battlefield intelligence, surveillance and target acquisition, strategic warning and defence. Over the years, the use of data and information fusion techniques has diversified tremendously and has extended to commercial and industrial sectors. Examples of non-military applications include condition-based maintenance, robotics, medical applications and environmental monitoring [122].

The Joint Directors of Laboratories data fusion model developed for the United States Department of Defense divides the multilevel data and information fusion process into low-level and high-level processes. The definitions of the functional levels of the model have been revised several times since it was first created about twenty years ago. Based on the current definitions, the low-level fusion process comprises Level 0 (data assessment) and Level 1 (object assessment), while the high-level fusion process consists of Level 2 (situation assessment), Level 3 (impact assessment), Level 4 (process refinement) and Level 5 (cognitive refinement). The aforementioned levels of fusion are briefly described below [123, 230].

Level 0: Data assessment

Data from sources such as sensors and databases are processed prior to fusion with other data at higher levels. Techniques include signal processing and other operations to prepare the data for subsequent fusion.

Level 1: Object assessment

Fusion of data that resulted from Level 0 processing to obtain estimates of the states (such as position, location, motion, attribute, characteristic or identity) of an entity (such as a spatially or geographically localized object or a fault condition in a mechanical system). Techniques include target tracking and pattern recognition.

Level 2: Situation assessment

Utilization of results from low-level fusion processes to evaluate the relationships (such as proximity, temporal relationship or communication among sources) among entities and their relationship (can be physical, organizational, informational or perceptual) to the environment (such as terrain, surrounding media or vegetation), as well as to aggregate the entities in time and space to derive an interpretation of the situation. Techniques are built from automated reasoning and artificial intelligence.

Level 3: Impact assessment

Inference/prediction about the effects of current evolving situation (events and activities derived at Level 2 process) on one's goals/objectives. Techniques utilized include automated reasoning, artificial intelligence, predictive modelling and statistical estimation.

Level 4: Process refinement (an element of Resource Management)

Utilization of data sources and tools for continuous monitoring to improve the real-time performance of the ongoing information collection/extraction and fusion processes.

Level 5: Cognitive refinement (an element of Knowledge Management)

Continuous monitoring of the ongoing interaction between the human user or decision maker and the data fusion system, with the aim of enhancing computer-aided cognition.

1.1 Research Objectives

In this thesis, we study some issues and applications of data and information fusion. The main research objectives are described as follows.

- Detailed survey on high-level information fusion:
  The focus of data and information fusion research is shifting from low-level information fusion towards high-level information fusion. We do a survey on problems and techniques related to high-level information fusion. It includes a review of several existing models for data and information fusion, as well as a discussion on application domains and topics for future research.

- Target tracking:
  With emphasis on manœuvring target tracking, we investigate the combinations of nonlinear filtering algorithms and interacting multiple model based filters for state estimation. Our aim is to obtain filtering algorithms that can achieve effective state estimation at moderate computational complexity.

- Intent inference:
  We develop an intent inference approach for military and civilian air traffic surveillance. Our aim is for the method to be able to provide accurate and timely inference, as well as to attain accurate and fast response/countermeasures against the subject being monitored.

1.2 Overview of Thesis

The main focus of data and information fusion research has previously been on low-level information fusion. The focus is currently shifting towards high-level information fusion. Compared to the increasingly mature field of low-level information fusion, the theoretical and practical challenges posed by high-level information fusion are more difficult to handle. Contributing factors include the lack of: well-defined spatiotemporal constraints on relevant evidence, well-defined ontological constraints on relevant evidence and suitable models for causality. In Chapter 2, some process models proposed for data and information fusion over the past few decades are reviewed. Based on the fusion levels of the current Joint Directors of Laboratories data fusion model, a detailed survey of existing literature and approaches for high-level information fusion is presented. Relevant application areas and topics with potential for further research are also discussed.

Chapter 3 deals with the topic of target tracking, an essential element of systems that perform tasks such as surveillance, navigation, aviation and obstacle avoidance. The emphasis of the discussion is placed on manœuvring target tracking. It is generally difficult to represent different behavioural aspects of the motion of a manœuvring target with a single model. Multiple model based approaches are useful for adaptive state estimation when tracking motion with variable behaviour. Therefore, these approaches are usually required when seeking solutions for manœuvring target tracking problems, which are generally nonlinear. In the recent years, new strategies have been developed via the combination of the Interacting Multiple Model (IMM) method and variants of particle filters. The former accounts for mode switching, while the latter account for nonlinearity and/or non-Gaussianity in the dynamic system models for the posed problems. Here, an IMM algorithm is considered for tracking target motion with manœuvres. The proposed algorithm comprises a constant velocity model, a constant acceleration model and a coordinated turn model. A variety of combinations of extended Kalman filters, unscented Kalman filters and particle filters are used for the models. The proposed algorithm is applied to three-dimensional (3D) manœuvring target tracking, as well as localization and tracking in a horizontal plane with the use of a Time Difference of Arrival (TDOA) system [61, 120]. In the simulation tests carried out, the results obtained show that superior performance in state estimation can be achieved at relatively modest computational costs, by using a computationally economical particle filter in the coordinated turn model, with extended Kalman filters and/or unscented Kalman filters in the remaining models.

The nonlinear filters and IMM algorithm variants are also applied to a problem on modelling financial option contract prices. Numerical tests are conducted using real data. The test results are analyzed to compare the performance of the individual filters.

Chapter 4 discusses intent inference, which involves the analysis of actions and activities of a target of interest to deduce its purpose. In an environment cluttered with many targets, loaded with information, and under stress, the human may not be able to perform well. Therefore, a cognitive aid that can derive possible intent inference and monitor the target may help augment human cognition and if possible, achieve better performance in intellectual tasks. Reports on the research done for two application problems are given. For the first problem, the objective is to determine the likelihood of weapon delivery by an attack aircraft under military surveillance. The second problem is concerned with conformance monitoring in air traffic control systems. The proposed solution is based on the analysis of flight profiles. Simulation tests are carried out on flight profiles generated using different combinations of flight parameters. In each simulation test, IMM-based state estimation is carried out to update the state vectors of the aircraft being monitored. Relevant variables of the filtered flight trajectory are subsequently used as inputs for a Mamdani-type fuzzy inference system (FIS) [152]. For the first application, the outputs produced by the FIS are the inferred possibilities of weapon delivery. For the second application, the FIS outputs are the inferred possibilities of non-conforming aircraft behaviour. The test results verify that the suggested method is practicable and provides timely inference that will aid human cognition and hence, assist critical decision making.

Next, we revert to the aforementioned problem on military surveillance. Taking into account constraints on computation time requirements, several IMM algorithm variants discussed in Chapter 3 are considered for the state estimation component of our proposed intent inference method. A comparison of the performance in state estimation is done for the filters. Subsequently, several issues pertaining to the extension of the proposed intent inference approach to handle approach by multiple aircraft are discussed.

Lastly, Chapter 5 gives a conclusion on this thesis and mentions some possible areas for further research.

1.3 Contributions of the Thesis

The following tasks are accomplished in this thesis.

- We have done an extensive survey of the existing literature and state-of-the-art approaches for high-level information fusion. Several application areas and topics of interest for exploration are highlighted, with relevant works from the research literature mentioned for reference.

- We have derived an algorithm for state estimation by combining the IMM method with extended Kalman filters, unscented Kalman filters and particle filters. The proposed algorithm consists of a constant velocity model, a constant acceleration model and a coordinated turn model. Different combinations of extended Kalman filters, unscented Kalman filters and particle filters have been used for the models. We apply the filtering algorithms to simulation problems on 2D and 3D manœuvring target tracking. The numerical results are analyzed via the comparison of state estimation errors, statistical analysis formulated as a hypothesis testing problem and comparison of state estimation errors with filter-calculated covariances. According to the test results obtained, IMM algorithm variants which use a computationally economical particle filter in the coordinated turn model, with extended Kalman filters and/or unscented Kalman filters in the remaining two models, show promise in attaining a balance between computational complexity and performance. They require relatively modest computational complexity and yield state estimation results that are comparable or superior to the other filtering algorithms implemented in the simulation tests.

We apply the above-mentioned filtering algorithms to a problem on modelling the prices of financial option contracts. Numerical tests are carried out using real data. The results are analyzed to assess the performance of the filters in state estimation.

- We have developed a new flight profile based approach for intent inference. The proposed fuzzy inference framework is applied to two problems, namely, flight mission of an attack aircraft and conformance monitoring in air traffic control/management. Experimental test results indicate that the suggested method is likely to provide timely and useful cognitive aid to decision makers in air defence and air traffic control/management.

  We consider several of the above-mentioned IMM algorithm variants for the state estimation component of the proposed intent inference approach. The estimation results are compared to identify additional suitable filters for state estimation in the proposed system.

Chapter 2

Survey of High-level Information Fusion

2.1 Introduction

Data and information fusion (DIF) involves a multifaceted, multilevel process of combining data from multiple sources, with the aim of acquiring information that is better (more useful and meaningful) than that would be derived from each of the sources individually (that is, without fusing). DIF is emerging as an important field of multidisciplinary study [77, 230]. This is due to increase in data and information flow, as well as improvement in communication, computing and sensor technology. The first applications of DIF techniques were in the military arena [122, 123, 321]. The use of DIF techniques for problem-solving has extended to many non-military applications in the commercial and industrial sectors [122, 123, 141, 161].

2.1.1 Review of Data Fusion Models

Over the last few decades, many process models have been proposed for DIF [123, 238]. Some of the data fusion (DF) models introduced over the years are briefly reviewed in the following subsections. More details on these models are found in the respective sources and the cited references therein.

2.1.2 Data Fusion Models Introduced in the 1980s

In the 1980s, the Intelligence Cycle [15, 105], the Boyd Control Loop [238, 257] and the Joint Directors of Laboratories data fusion (JDL DF) model [39, 121, 205, 300, 304] were developed.

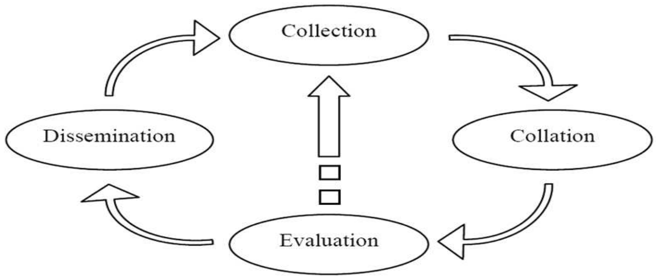

Figure 2.1: The Intelligence Cycle [15].

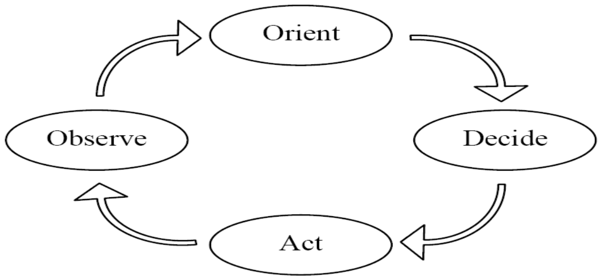

Figure 2.2: The OODA Loop [238].

2.1.2.1 The Intelligence Cycle

In the Intelligence Cycle, the intelligence process is described as a cycle applicable for modelling the data fusion process. This model consists of four phases (shown in Figure 2.1): *collection* (deployment of assets such as electronic sensors or human derived sources to obtain raw intelligence data, which is usually presented in the form of an intelligence report with a high abstraction level); *collation* (analysis, comparison and correlation of associated intelligence reports); *evaluation* (fusion and analysis of collated intelligence reports) and *dissemination* (distribution of the fused intelligence to users who use the information for decision making).

2.1.2.2 The Boyd Control Loop

The Boyd Control Loop, also known as the Observe, Orient, Decide, and Act (OODA) Loop, was first proposed to model the military command and control (C2) process. It comprises four phases (shown in Figure 2.2): *Observe* (gather information from the environment); *Orient* (gain situation awareness and perform situation/threat assessment based on the information gathered); *Decide* (respond to situation and work out follow-up actions) and *Act* (execute the planned response/action). The emphasis is placed on shortening the cycle to perform the Observe to Act loop, to the extent that the opponent cannot respond in time to carry out countermeasure, thus gaining superiority

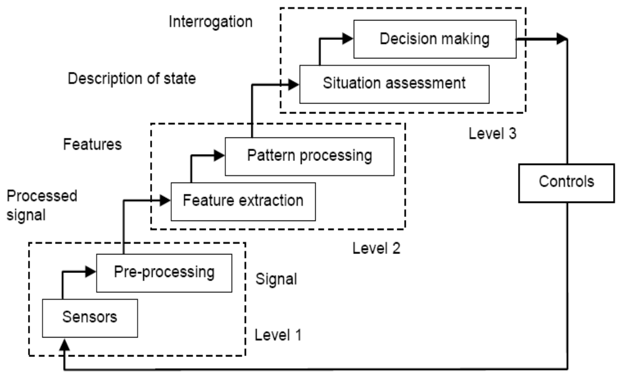

Figure 2.3: The Waterfall model [96].

in the battlespace. This model is well received by military commanders and decision makers.

The commonly used JDL DF model was proposed for categorizing data fusion related functions. A detailed discussion on this model is given in Section 2.2.

2.1.3 Data Fusion Models Introduced in the 1990s

During the 1990s, the Waterfall model [15, 96, 105], the Dasarathy model [74, 75], the Visual Data-Fusion (VDF) model [39], the Omnibus model [15] and the Endsley model [39, 92, 93] were proposed.

2.1.3.1 The Waterfall Model

The Waterfall model consists of three levels of representation (shown in Figure 2.3):

- Level 1 (sensing, signal processing) -
  proper transformation of raw data is carried out to provide necessary information about the surroundings, via the use of models (based on experimental analysis or on physical laws) of the sensors and where possible, of the measured phenomena;

- Level 2 (feature extraction, pattern processing) -
  with the aim of minimizing the data content and maximizing the information delivered, feature extraction and fusion are done to produce a list of estimates and their associated probabilities (and beliefs), which provide a symbolic level of inference about the data;

- Level 3 (situation assessment, decision making) -
  relationships are established between objects and events; based on the repository

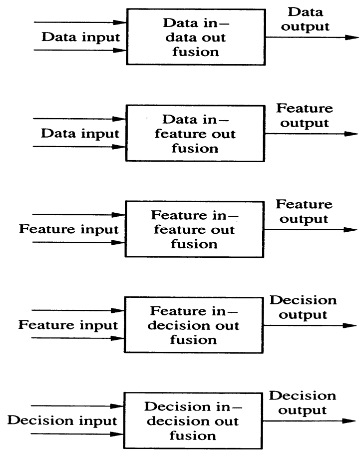

Figure 2.4: The Dasarathy model [74].

of information available and the human interaction, possible routes of action are assembled.

The focus is on the processing functions at the lower levels. The lack of explicit depiction of the feedback appears to be the major limitation of this model.

2.1.3.2 The Dasarathy Model

The DF process has been commonly identified as a hierarchy with three general levels of abstraction: *data* (more specifically, sensor data), *features* (intermediate-level information) and *decisions* (symbols or belief values). Dasarathy [74, 75] pointed out that fusion may occur both within and across these levels. The Dasarathy model was proposed to expand the preceding hierarchy of fusion into five categories of input-output based fusion (corresponding analogues stated within parentheses): *Data In-Data Out* fusion (data-level fusion); *Data In-Feature Out* fusion (feature selection and feature extraction); *Feature In-Feature Out* fusion (feature-level fusion); *Feature In-Decision Out* fusion (pattern recognition and pattern processing) and *Decision In-Decision Out* fusion (decision-level fusion). This model is based on DF functions (illustrated in Figure 2.4) instead of tasks and may be incorporated in each of the fusion activities.

2.1.3.3 The Visual Data-Fusion Model

The Visual Data-Fusion model (see Figure 2.5) was proposed as an extension of the JDL DF model, with a human participant added integrally. It has the following advantages [39]:

- maximization of relevant information with minimal display of information;

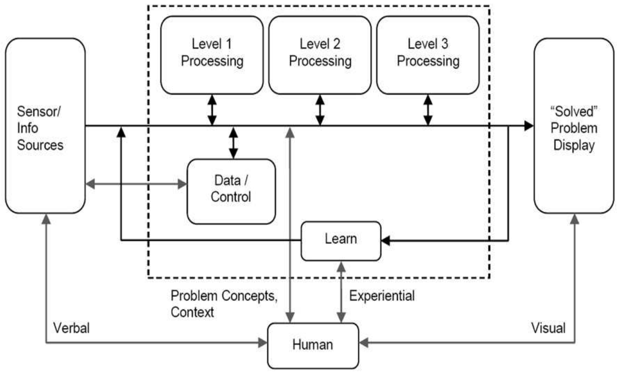

Figure 2.5: The Visual Data-Fusion model [39].

- ability to provide increasingly sophisticated problem queries, in addition to tailor information fusion (IF) system capabilities for use by all skill levels of users;
- problem-driven system that relates to user's needs directly, through response to his personal perception of the problem situation.

The following premises are embodied in the VDF model [39]:

- the human is a central participant in information fusion, a creative problem-solving process;
- information derived from the fusion process that is visualized by the human is primarily used to help him gain fuller perception, as well as possible approaches towards solving the problem;
- imagery is used as the perceptual transport for user visualization, in order to minimize the amount of information required by the human to solve the problem.

Basic VDF models are used as building-block elements for visual situation awareness and distributed VDF processes. More details on these research topics can be found in [39].

2.1.3.4 The Omnibus Model

The Omnibus model was proposed as a unification of the Intelligence Cycle, the JDL DF model, the OODA Loop, the Dasarathy model and the Waterfall model. Properties of this model include: explicit feedback; acknowledgement of the *loop within loop* concept; retention of the general structure of the OODA Loop; incorporation of the fidelity of representation expressed by the Waterfall model into each of its four main modules and

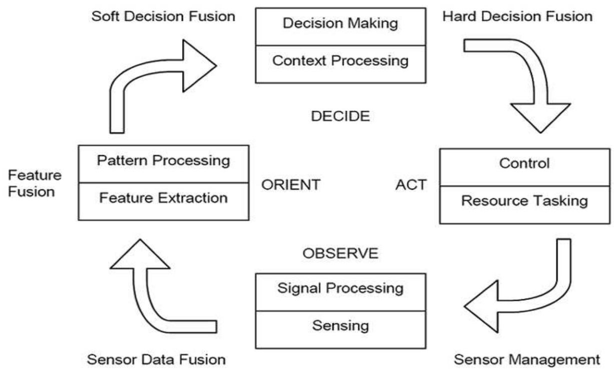

Figure 2.6: The Omnibus model [123].

explicit indication of points in the processes where fusion may take place. Figure 2.6 presents the layout of this model.

The Endsley model is widely used for modelling situation awareness. Section 2.3.1 provides elaboration on this model.

2.1.4 Data Fusion Models Introduced in the 2000s

The following data fusion models have been proposed in the first half of this decade:

- the Object-Centered information fusion model [166],
- the Extended OODA model [291],
- the Transformation of Requirements for the Information Process (TRIP) model [123],
- the Unified data fusion (λJDL) model [39, 184],
- the Dynamic OODA Loop [42],
- the JDL-User model [28].

2.1.4.1 The Object-Centered Information Fusion Model

Kokar et al. introduced a fusion process reference model based on object-oriented design principles. The proposed model addressed essential issues on the design of data fusion systems with a top-down approach. Formal methods were adopted for model analysis at the design stage. They also discussed the need to develop psychological theories related to human-computer interaction (HCI). Research in this area was required for facilitating

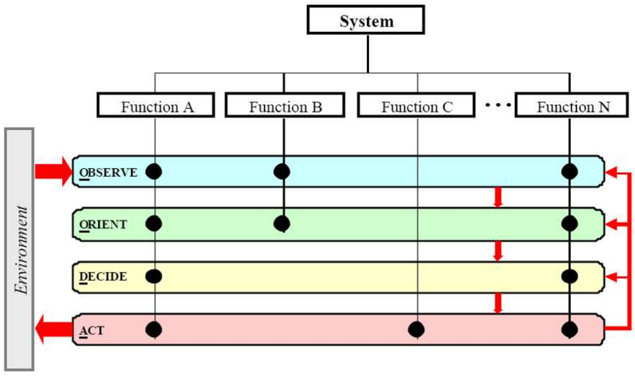

Figure 2.7: The Extended OODA model [291].

the proper integration of human and computer objects by fusion system designs based on the proposed object-oriented model.

2.1.4.2 The Extended OODA Model

Shahbazian et al. [291] proposed the Extended OODA model which enables multiple concurrent and potentially interacting data fusion processes. This model can be applied to obtain a high-level functional decomposition of a system that uses data fusion for decision making. Each high-level function is examined in terms of the OODA decision loop and can be further decomposed and evaluated with respect to each OODA phase.

The Extended OODA model (see Figure 2.7) has some properties that are consistent with those of several preceding models (stated within parentheses): closes the loop between the decision making and its surroundings (OODA Loop); has increasing level of abstraction for information processing in each level (JDL DF model) and provides the *loop within loop* capability (Omnibus model).

2.1.4.3 The TRIP Model

The TRIP model was developed with the purpose of understanding a tactical commander's transformation of information needs to task assignment of sensor resources. The developers stated the following goals that they aimed to accomplish with this model [123]:

- describe the process for developing collection tasks from information requirements;
- understand relationships between collection management and the situation estimation process;
- understand where the *human in the loop* is required;

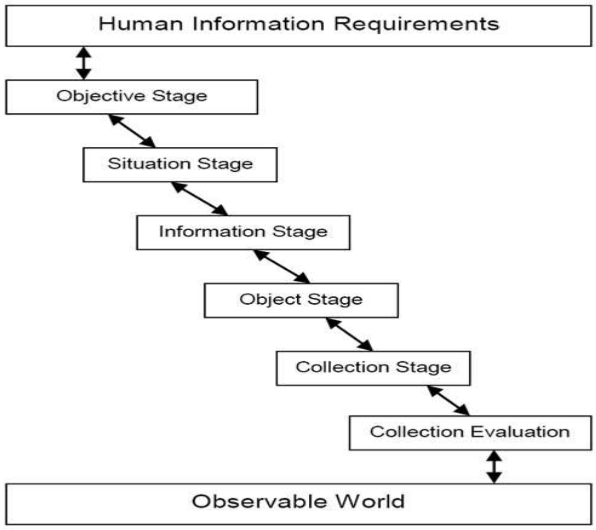

Figure 2.8: The TRIP model [123].

- understand the internal and external drivers for the intelligence, surveillance, and reconnaissance process.

Identification of processing functions and the detailed information interfaces between them was attempted. A link between human information requirements and data collection was provided by this model (depicted in Figure 2.8).

2.1.4.4 The Unified Data Fusion (λJDL) Model

The λJDL model (also known as the deconstructed JDL DF model), a revision of the JDL DF model (the version proposed in [304]), used the following definitions for its fusion levels (see Figure 2.9):

- Level 1 (identification of objects from their properties) -
  object fusion: process of utilizing one or more data sources over time to assemble a representation of objects of interest in an environment;
  object assessment: stored representation of objects obtained through object fusion;

- Level 2 (identification of relations between these objects) -
  situation fusion: process of utilizing one or more data sources over time to assemble a representation of relations of interest between objects of interest in an environment;
  situation assessment: stored representation of relations between objects obtained through situation fusion;

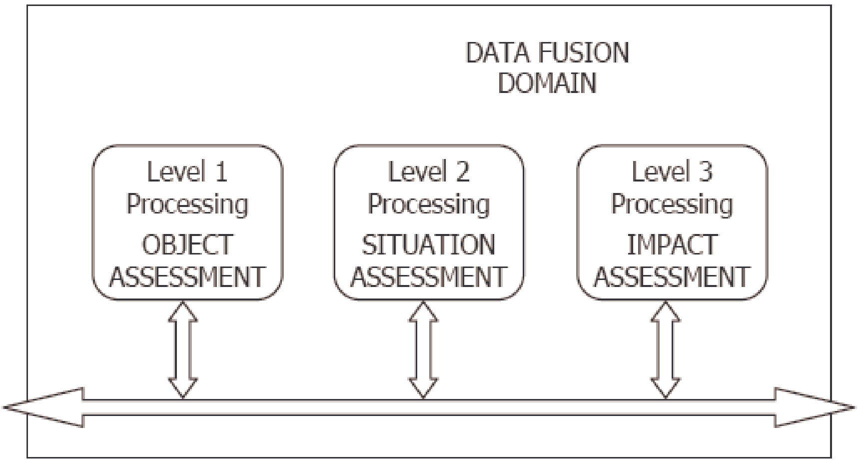

Figure 2.9: The λJDL model [184].

- Level 3 (identification of the effects of these relationships between these objects) - *impact fusion*: process of utilizing one or more data sources over time to assemble a representation of effects of situations in an environment, relative to user intentions; *impact assessment*: stored representation of effects of situations obtained through impact fusion.

The model was proposed for the development of a data fusion system for fusing three distinct types of processes that involved both humans and machines:

- psychological processes (human-related),
- technological processes (machine-related),
- integration processes (interaction between the psychological and technological processes).

The model could be applied to different aspects of the data fusion problem, depending on the different interpretations of the model components (object, situation, impact) obtained from the different combinations of the above processes.

2.1.4.5 The Dynamic OODA Loop

There exist criticisms that the OODA Loop fails to capture the dynamic nature of decision making in the military command and control process, as it has a limited focus on faster decisions. The Dynamic OODA Loop (shown in Figure 2.10) was proposed as a generic model of military command and control, based on concepts from the OODA Loop and cybernetic models of C2.

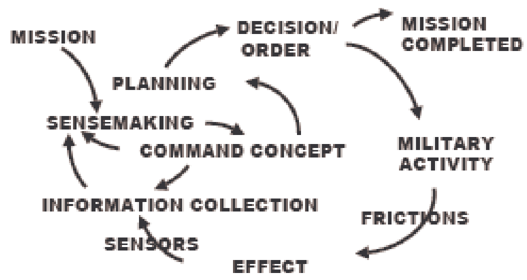

Figure 2.10: The Dynamic OODA Loop [42].

This model provides the identification of functions essential for effective C2. The problem of handling delays in C2, a form of dynamic decision making, is also dealt with. The required functions are: *sensemaking* (understanding of the current mission/situation in terms of what can be done); *command concept* (commander's overall concept of the operation); *planning* (translation of the command concept into decisions/orders); *information collection* (guided by the command concept) and *decision* (commitment to a course of action (COA)).

Other modifications of the OODA Loop include the M-OODA Loop [279] and the C-OODA Loop [43].

Discussion on the JDL-User model, which was proposed to extend the JDL DF model to support a *human-in-the-loop* decision process, is deferred to Section 2.6.

The remainder of this chapter is as follows. Section 2.2 is focussed on the JDL DF model, which has been revised and extended several times since it was first proposed. Sections 2.3 to 2.6 discuss the higher levels of fusion in the JDL DF model and some existing literature pertaining to the respective levels. Section 2.7 presents some application areas of high-level information fusion. In Section 2.8, summarizing remarks are made and potential topics for further research are considered.

2.2 The JDL Data Fusion Model

The original JDL DF model (shown in Figure 2.11)[1] was created by the JDL Data Fusion Group of the United States Department of Defense [121]. It is a functional model developed with the aim of facilitating communication, comprehension, coordination and cooperation among diverse DF communities to identify and solve problems to which DF can be applied.

The first revision of the initial JDL DF model was proposed by Steinberg et al. [304].

[1]Definitions corresponding to the symbols in the figures in this chapter are in the List of Acronyms.

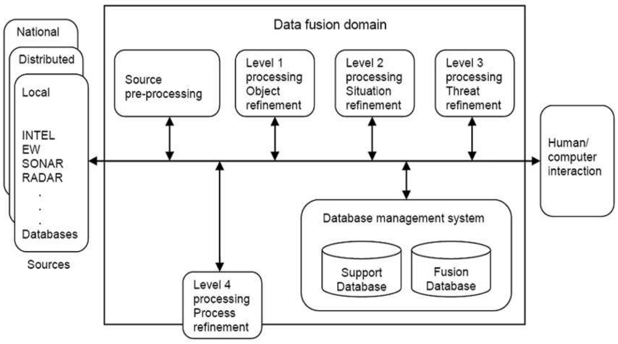

Figure 2.11: JDL DF model [121].

They broadened the definitions of fusion concepts and functions beyond the original focus on military and intelligence problems, as well as described the need for an approach to the standardization of an engineering design methodology for fusion processes. They also proposed to refine definitions for the fusion "levels" characterized in the original JDL DF model as follows [304]:

- Level 0 (Sub-Object Data Assessment) -
  estimation and prediction of observable states of signals or features;

- Level 1 (Object Assessment) -
  estimation and prediction of entity states based on data association, as well as continuous and discrete state estimation;

- Level 2 (Situation Assessment) -
  estimation and prediction of relationships among entities;

- Level 3 (Impact Assessment) -
  estimation and prediction of effects of entities' actions on goals/missions;

- Level 4 (Process Refinement) -
  an element of Resource Management (RM) that encompasses adaptivity in the data collection and fusion processes to support mission objectives.

Figure 2.12 shows this revised version of the JDL DF model, which included the introduction of a "Level 0" to the original model. The five fusion levels were categorized into the low-level fusion process (Levels 0 and 1) and the high-level fusion process (Levels 2 to 4) [163, 230].

The JDL DF model accounts for automatic machine processing, but not for human processing. To address issues related to extending the human capabilities within the

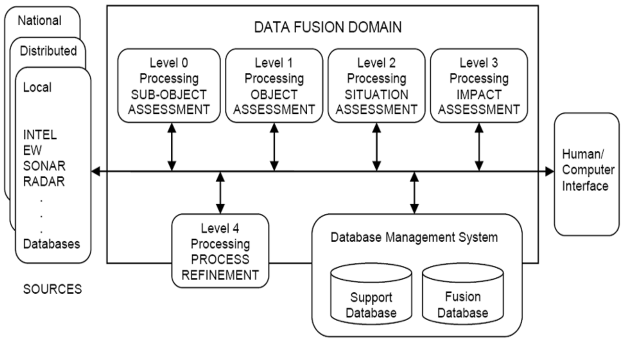

Figure 2.12: Revised JDL DF model [304].

fusion process, the concept of Level 5 data fusion process was first introduced by Hall et al. [125] and subsequently, in an independent work by Blasch and Plano [28]. In both works, the authors asserted the need to acknowledge functions necessary for supporting a human-in-the-loop decision process. More details on Level 5 processing are discussed in Section 2.6.

More recently, another revision to the JDL DF model (illustrated in Figure 2.13) was suggested by Llinas et al. [205, 300]. The refinement involved a re-examination of the JDL DF level structure. The data fusion levels were extended to a newly introduced set of dual resource management levels (encompassed functions include signal/signature management, individual RM, coordinated RM, goal management and system engineering). Based on the entities of interest to information users, revision of the definitions for data fusion functional levels were suggested as follows [205, 300]:

- Level 0 (Signal/Feature Assessment) -
  estimation and prediction of states of signals or features;

- Level 1 (Entity Assessment) -
  estimation and prediction of parametric and attributive states of entities;

- Level 2 (Situation Assessment) -
  estimation and prediction of relational/situational states of entities;

- Level 3 (Impact Assessment) -
  estimation and prediction of effects of fused entity/situation states on mission objectives;

- Level 4 (Performance Assessment) -
  estimation and prediction of a system's measures of performance and measures of effectiveness based on given desired system states and/or responses.

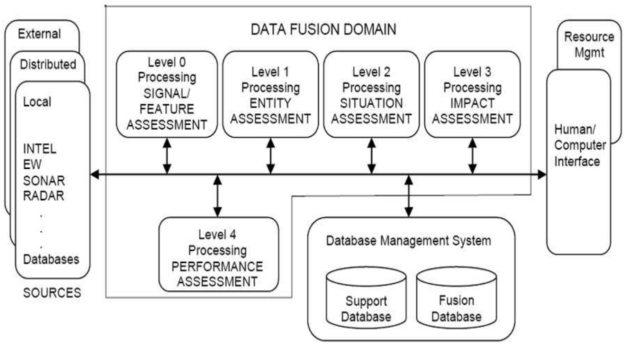

Figure 2.13: Revised JDL DF model [205].

In the revised version of the JDL DF model [205], the previous Level 4 (Process Refinement) function [304] was categorized as being within the Resource Management model levels, while the proposed Level 5 [28,125] was subsumed as an element of Knowledge Management within Resource Management. The reason was that incorporation of a Level 5 into the JDL DF model had then not achieved common usage or acceptance by the fusion community.

A further upgrade/revision to the JDL DF model (see Figure 2.14) was assessed by the Data Fusion Information Group (DFIG) [24,31]. The aim was to separate the information fusion and management functions. A detailed explanation on the model can be found in [25]. The definitions for this model, based on the version of the JDL DF model proposed in [304], are:

- Level 0 (Data Assessment) -
  estimation and prediction of observable states of signals or features;

- Level 1 (Object Assessment) -
  estimation and prediction of entity states based on data association, as well as continuous and discrete state estimation;

- Level 2 (Situation Assessment) -
  estimation and prediction of relationships among entities;

- Level 3 (Impact Assessment) -
  estimation and prediction of effects of entities' actions on goals/missions;

- Level 4 (Process Refinement) -
  an element of Resource Management that encompasses adaptivity in the data collection and fusion processes to support mission objectives;

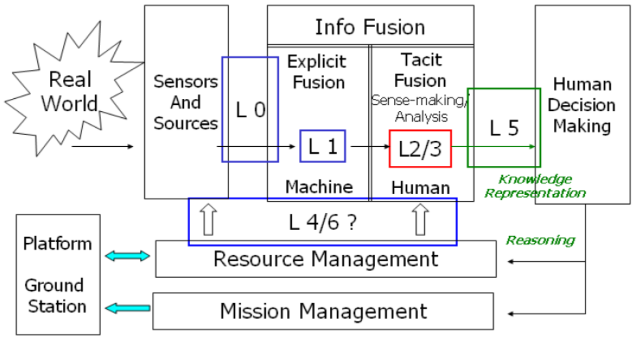

Figure 2.14: DFIG 2004 model [24, 31].

- Level 5 (User Refinement) -

  an element of Knowledge Management that encompasses adaptivity in the determination of user query and access to information, as well as adaptivity in the retrieval and display of data, to support cognitive decision making and actions;

- Level 6 (Mission Management) -

  an element of Platform Management that encompasses adaptivity in the determination of spatial-temporal asset control, as well as route planning and goal determination to support team decision making and actions.

Other recent revisions of the JDL DF model include the State Transition Data Fusion (STDF) model [185–187] and the ProFusion2 (PF2) model [258].

After many years of intensive research, low-level fusion is becoming an increasingly mature field. The research focus is currently shifting towards fusion at higher levels. The significant amount of interest in high-level information fusion is evident in panel discussion sessions being dedicated to address issues related to this field at the International Conference on Information Fusion held in the recent years:

- 2004 - Challenges in Higher Level Fusion: Unsolved, Difficult, and Misunderstood Problems/Approaches in Levels 2-4 Fusion Research,

- 2005 - Issues and Challenges of Knowledge Representation and Reasoning Methods in Situation Assessment (Level 2 Fusion) [27],

- 2006 - Issues and Challenges in Resource Management and Its Interaction with Level 2/3 Fusion with Applications to Real-World Problems [26],

- 2007 - Results from Levels 2/3 Fusion Implementations: Issues, Challenges, Retrospectives and Perspectives for the Future,

- 2008 - High-level Information Fusion: Challenges to the Academic Community.

The journal Information Fusion has also published a special issue on high-level information fusion and situation awareness [167, 170, 187, 201, 223, 254, 313, 338].

Research and development of techniques in high-level IF are being actively carried out in various application domains. With inspiration from a military process, Sycara et al. [313] developed a computational framework, High-level Information Fusion Environment (HiLIFE), to implement a novel integrated conceptual architecture for higher levels of fusion. Karlsson [161] investigated dependability requirements and uncertainty management methods in generic high-level IF. The subsequent sections review some work on high-level IF in the existing literature.

2.3 Situation Awareness

Level 2 fusion, also known as *Situation Assessment* (SA), is concerned with the determination and interpretation of relationships among objects. The objectives at this level include the derivation of high-level inference and the identification of meaningful events and activities [230]. *Situation Awareness* (SAW) involves the identification and monitoring of various relationships among Level 1 physical and abstract entities, as well as various relations among them [286]. SA is regarded as the process of achieving, acquiring or maintaining SAW. SAW is commonly modelled with the Endsley model [92, 93] described in the next subsection.

2.3.1 Endsley's Situation Awareness Model

Endsley's SAW model (shown in Figure 2.15) uses a general definition of SAW that is applicable across many domains: "Situation awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future". The three hierarchical phases of the definition are [92, 93]:

- Level 1 SAW (*Perception* of the elements in the environment) -
  perceive status, attributes and dynamics of relevant elements in the environment;

- Level 2 SAW (*Comprehension* of the current situation) -
  based on a synthesis of disjoint Level 1 elements, includes perceiving and attending to information, as well as integrating multiple pieces of information and a determination of their relevance to the operator goals;

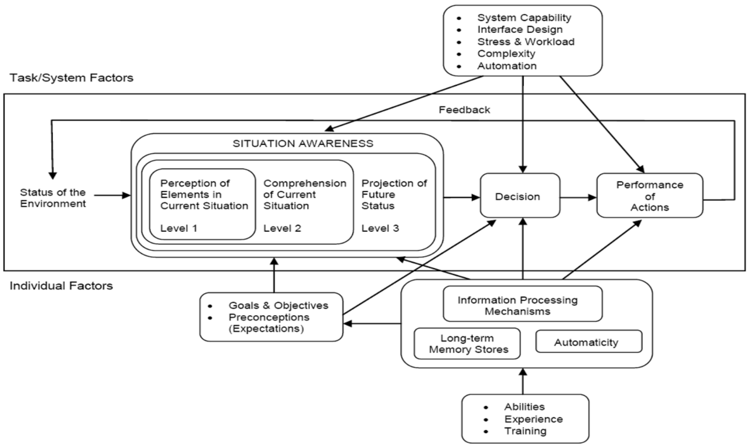

Figure 2.15: Endsley's SAW model [92, 93].

- Level 3 SAW (*Projection* of future status) -
  ability to forecast/anticipate future situation events and dynamics, which is achieved through knowledge of status and dynamics of the elements and comprehension of the situation (both Levels 1 and 2 SAW), allows for timely decision making.

2.3.2 Issues and Approaches

General issues and challenges in situation assessment and situation awareness have been addressed by different researchers with various perspectives and approaches [27].

- Kokar [164, 165] identified and discussed problems pertaining to automatic SA/SAW. Approaches for solving these identified problems were proposed and compared.

- Salerno et al. [286] explored various techniques believed to be necessary for providing SAW. They also investigated how those techniques could be bound together to form an overall system architecture, as well as how various sources of information contributed to the problem of maintaining constant awareness of the environment one was in.

- A detailed discussion on developing a conceptual framework for SA and SAW was given by Salerno [285]. He also addressed issues and perspectives on high-level information fusion processing.

- Kadar [159] addressed issues in SA and associated Knowledge Representation and Reasoning models, with focus on a human perceptual reasoning-based model framework.

- Gorodetsky et al. [117] did an analysis of formal frameworks proposed for specification of the situation models. Their focus was on approaches and algorithms for on-line update of SA, on the generic architecture of the SA systems.

- Qureshi and Urlings [264] proposed an operator assistant with a flexible concept of automation, with the objective of enhancing SAW.

- Smart et al. [296] investigated knowledge-based approaches to improving SAW in humanitarian operational deployment. A tool for intelligent information fusion, Technical Demonstrator System, was developed for the SAW enhancement task. A functional overview of the system with respect to several capability areas was presented.

Many techniques and tools have been developed for solving Level 2 fusion problems. The rest of this section will discuss some of the research work available in the literature.

In Level 2 fusion, the concept of data association is much broader than in the Level 1 case. Transition from Level 1 to Level 2 fusion brings about tremendous increase in computational complexity. Kokar et al. [169] studied the issue of data association, as well as investigated the use of ontologies [171, 172, 216–218] in alleviating the problems of computational complexity in processing, for Level 2 fusion. Mathematical metrics for data association were introduced by Stubberud et al. [307] to develop an automated Level 2 fusion approach. These measures were developed with mathematical rigour. In [308, 309], target uncertainty information was incorporated into the formulation of the metrics proposed in [307] and the augmented metrics were applied to battlespace problems.

Transition from Object Assessment to Situation Assessment is accompanied with a shift from numeric representation to symbolic representation. For SA, it is necessary to deal with issues related to the determination of the symbols to be used and the assignment of meaning to the chosen symbols. Nowak and Lambert [242] discussed the use of ontologies for specifying formal theories with logic to assign meaning to symbols. Frameworks for using ontologies and agents in semantic fusion of legacy information sources were also presented.

Das et al. [72] developed an approach of using Bayesian belief networks (BBNs) for situation assessment and fusion of information. Their objective was to process and correlate data from multiple sources to generate an accurate and timely picture of the battlespace in order to achieve information dominance.

Das and Lawless [73] presented a network-based truth maintenance system that used BBNs for fundamental enabling of its truth maintenance mechanism. For Level 2 fusion, the BBNs were integrated with probabilistic integrity constraints for inconsistency detection. Computer tests were carried out for the proposed system on a military command and control platform.

Baclawski et al. [11] proposed a formal basis for situation awareness using a formal method system and an ontology language. The architecture made use of sources, as well as multidisciplinary techniques from logic, human-computer interaction and data fusion. Another objective of the project was to show ways to convey relevant symbolic information to a situation awareness system and possible inference output based on this input. Versatile Information Systems, Inc. developed a tool, Situation Awareness Assistant (SAWA), to monitor the evolution of higher-order relations within a situation. Formal and generic reasoning techniques (combination of logical inference with Bayesian belief propagation) for Level 2 fusion were used [219]. It consisted of a collection of flexible ontology-based IF tools required for identifying and tracking user-defined relations. The purpose of SAWA was to allow offline development of problem specific domain knowledge, before applying it at runtime to fuse and analyze Level 1 data. In [220], advantages and limitations of certain approaches, techniques and technologies applied to situation awareness during the process of developing SAWA were reported. Related results can be found in [168, 217, 218].

A set of characteristics identified for a tactical situation assessment problem was highlighted by Powell [260]. Associated knowledge and computational requirements for representation and reasoning were also discussed. In the context of attack helicopter missions, Entin [94] investigated methods for displaying dynamic tactical information to maintain high levels of situation awareness and task performance. Fountain and Drager [103] developed a JDL Level 2 fusion and exploitation architecture to support accurate and comprehensive military situation assessment in real-time. Royer and Challine [282] presented a generic and extensible prototype platform for intelligence fusion. Generic and modular fusion support services in situation assessment were developed as part of the project. Goossens et al. [115] reported the development of support functionality for an unmanned aerial vehicle operator station in order to obtain Level 3 situation awareness. The design requirements for the functionality and results obtained from implementing the developed functions in a simulation scenario were discussed. Blackman and Popoli [22, Chapter 12] focussed on fighter air-to-air tactical intercepts in their discussion on SA. They explained in depth the important role of SA in tracking system process. They also introduced a system architecture for SA.

A peer-to-peer situation awareness system was developed by Hinchion et al. [136] to address challenges within a dynamic network-centric battlefield. The objective was to provide the information infrastructure to enable information superiority for the defence forces.

Oxenham et al. [248, 249] investigated automatic target identification via the integration of distributed data fusion (relevant to network-centric warfare (NCW)) and the fusion of disparate types of uncertain data (relevant to interoperability). In particular, target identity estimates were generated by local heterogeneous data fusion systems in a network and then fused using newly developed Bayesian theory based distributed target identification algorithms.

Landis et al. [188] established an information fusion process in SAW using multiple arrays of sensory data. They designed two different template matching algorithms for classifying military convoys. Simulation test results obtained from the implementation of the two convoy aggregate identification algorithms were compared.

Shu and Kazuo [294] provided a conceptual and theoretical framework for inferring team SAW in cooperative activities. The principle to assess appropriateness of team SAW on the developed framework was also proposed.

Salmon et al. [287] reviewed existing SAW measurement techniques in order to determine if they were suitable for use in command, control, communications, computers and intelligence (C4I) environments. The authors made the recommendation that a multiple-measure (or toolkit) approach which utilized different measurement techniques be used to measure SAW in C4I environments.

2.4 Impact Assessment

Level 3 fusion, formerly known as *Threat Assessment* (TA) in the original JDL DF model, was redefined as *Impact Assessment* to accommodate expansion in the concept of Level 3 fusion [304]. Impact Assessment deals with the determination of the effect of current situational states on user objectives. It involves the prediction of the intent (alternative courses of action) for entities, as well as the estimation of the degree or severity with which impending (possibly adversarial) events may occur.

Attempts to achieve effective Level 3 fusion have been made via the development of techniques based on a wide variety of disciplines. Some of the existing research work will be presented in the remainder of this section.

Dall [69] outlined a decision aid to provide automatic situation and threat advice in the Air Defence Ground Environment (ADGE). An application centric approach was adopted. The application was considered in terms of the required outputs (outcomes from threat assessment), inputs (data) and domain knowledge. The characteristics of the ADGE served as a basis for the proposed approach, which had the interesting consequence that threat assessment was made without explicitly performing situation assessment. It was only when there was a need to support a particular threat hypothesis that relationships between entities were estimated.

Steinberg [301, 302] developed concepts for a systematic approach to characterize, predict, recognize and analyze threat situations and activities, with the goal of automation for some of the functions. His focus was on cases in which malicious agents existed. The work was based on advances in Situation, Ontology and Estimation theory.

Benavoli et al. [19] developed an information fusion system to provide TA in the context of evidential networks. The system was proposed to perform defence and security tasks in support of decision making by a commander.

Berrached et al. [20] introduced a model to illustrate the application of a fuzzy relational equation algorithm to threat analysis in the context of Computer Generated Forces systems. The proposed algorithm generated data from precedent information and runs, yielding new outcomes of the algorithm that were more realistic and more accurate than earlier ones.

Qu and He [263] presented a method for TA to tactical command and control system. Their approach was based on multiple attribute decision making and fuzzy set theory was applied to it.

Huang [142] proposed the application of real-time automated decision making techniques for airborne early warning purposes. The approach was based on modelling the operator's decision process. Implementation involved integrating a TA module and a tactical planning module, as well as testing the systems in an advanced simulation facility. Results obtained from the simulation tests were presented.

Nguyen [236] gave a report on a research program on target TA comprising two components, namely, capability and intent assessment. TA was analyzed using Cognitive Work Domain Analysis technique. A model of intent assessment based upon Bayesian networks was applied to assess target intent from uncertain and incomplete evidence.

Okello and Thorns [244] presented a Bayes net based TA for air defence. To evaluate the threat posed by a given intruder aircraft on a specified asset, the proposed approach made use of target state estimates and their measures of uncertainty from a tracking and data fusion module.

An and Liu [7] presented a method which applied different neural networks to evaluate the threat degree according to the fuzzy decision making rule. This resulted in a new threat judgement algorithm that could give more reasonable result.

A modified probabilistic neural network that could achieve reliable assessment of abnormal or suspicious behaviours in an automated visual surveillance application was proposed by Jan [151]. The model required significantly reduced computation compared to other artificial neural network models with comparable classification performance for real-time video processing.

Threat classification has a direct impact on the decisions made in order to eliminate the threats. To stabilize this classification, Allouche [6] proposed that the threat stabilization problem be tackled by using a neural approach based on Kohonen's self-organizing maps. Some important features from the kinematics of the moving entity were extracted and fused. Feasibility of the method was verified by the simulation results obtained.

The determination, prediction and analysis of potential enemy courses of action are key elements of threat assessment data fusion processing. Charles River Analytics, Inc. designed tools for these crucial tasks in threat assessment. An Intelligent Threat Assessment Processor (ITAP) for enhancing tactical threat assessment was developed in [113]. The major components of ITAP included a genetic algorithm-based approach to predict enemy COAs and a fuzzy logic-based analysis of predicted enemy COAs to infer enemy intent and objectives. It also provided necessary functionality to support multilevel data fusion, in conjunction with an Intelligent Fusion and Asset Management Processor being developed. In [112], an architecture that employed genetic algorithms to generate and evaluate enemy COAs was formulated. The output of this architecture was a set of enemy COAs that could be evaluated based on their potential effectiveness against friendly forces.

Paradis et al. [251] discussed threat evaluation and weapons allocation (TEWA) in the context of network-centric warfare. A decision based framework for performing TEWA was developed through applying the applied cognitive work analysis methodology. There was also an outline of emerging concepts in TEWA, as well as a discussion on the applicability of these concepts to NCW.

2.4.1 More on Fusion at Levels 2 and 3

According to [304], Level 3 fusion can be perceived as a subset of Level 2 fusion due to the broad definition for the latter. Assignment at Level 3 is usually inferred from Level 2 associations, although processing at the fusion levels need not be performed in order [205]. In addition, given corresponding inputs, any one level can be processed on its own. In this subsection, some works that have tackled both Level 2 and Level 3 fusion issues are briefly discussed [52]. More information can be found in the references and those cited therein.

Petterson et al. [253] proposed to approach the situation/threat assessment (STA) problem for air combat application with a temporal representation, in contrast to the conventional approach with a spatial representation. The approach involved the transformation of a three-dimensional spatial problem into a temporal representation, which was used as a base for further processing.

Johnson and Dall [157] developed a system to provide the air force with an intelligent tool to streamline the tasks of performing STA within the battlespace environment.

Oxenham [247] discussed situation awareness enhancement with emphasis on the role of threat analysis (determination and evaluation of threats). An extension to an existing methodology for extracting air target behaviours from sensor track data was proposed and applied to the problem of recognizing and ranking threats.

Hinman [137] carried out investigation on some approaches to significantly improve situation assessment and threat prediction.

A framework of knowledge-based system that was suitable for military applications was proposed by Choi et al. [65]. They described the system as a near real-time automation system for intelligence preparation of the battlefield, an important role to assess situation and threat missions in military operation.

Looney and Liang [208] developed an integrated multi-phase approach to higher level data fusion. Application to a simple simulation example demonstrated the process which involved data clustering, followed by STA.

Bomberger et al. [37] presented an approach to higher-level information fusion problems of knowledge representation and learning. The proposed approach combined elements from the fields of neural modelling and artificial intelligence (including synchro-

Application domain	Approach/Technique	Reference
Automated visual surveillance	Modified probabilistic neural network	[151]
Data association	Ontology	[169, 171, 172, 216–218]
	Mathematics based metrics	[307–309]
Semantic fusion	Ontology	[242]
Tactical defence	Kohonen's self-organizing maps	[6]
- Air defence	Neural networks	[7, 137]
- C4ISR	Evidential networks	[19]
- Enemy courses of action	Fuzzy logic/Fuzzy set theory	[20, 102, 137, 208, 234, 253, 263]
- Ground battlespace	Knowledge-based approaches	[37, 65, 137]
- Information warfare	Axiomatic approach	[69]
- Interoperability	Bayesian inference/network/theory	[72, 73, 137, 208, 236, 244, 248, 249]
- Maintenance of consistency in	Genetic algorithms	[112, 113, 137]
intelligence database	Self-organizing peer-to-peer SAW	[136]
- NBD/NCW	system	
- Threat analysis	Real-time automated rule-based	[142]
- Threat stabilization	system	
	Situation, ontology, estimation theory	[157, 301, 302]
	Geometry, contextual information,	[247]
	target behaviour extraction	
	Cognitive system engineering	[251]
	Information theory	[253]
	Multiple attribute decision making	[263]
	Ontology	[282]
	SAW measurement techniques	[287]

Table 2.1: Situation and impact assessment - issues and approaches.

nization within spiking neural networks), spike timing-based associative learning and semantic knowledge networks.

Ng et al. [102, 234] proposed a fuzzy inference approach based on the analysis of aircraft flight profiles. The method utilized available knowledge on the preceding activities of a target of interest to predict its future action. The approach was implemented on two application problems, namely, military surveillance and conformance monitoring. Chapter 4 provides a detailed discussion extended from this work.

Table 2.1 gives a summary of the problems and techniques on situation and impact assessment discussed in Sections 2.3 and 2.4.

2.5 Process Refinement

Level 4 fusion was known as *Process Refinement* in the earlier versions of the JDL DF model [304]. The process involved resource management to improve the results obtained at the lower levels of data fusion [230]. In the recent revision of the JDL DF model [205], the data fusion levels were extended to their dual resource management levels. In addition, a new Level 4 of data fusion and its corresponding dual Level 4 of resource management were introduced. A redefinition Level 4 (*Performance Assessment* (PA), also known as *Performance Evaluation* (PE)) was proposed with the previous Level 4 (Process Refinement) function [304] categorized as being within the resource management model levels. Based on a given desired set of system states and/or responses, the Level 4 data fusion functions combined information to estimate a system's measures of performances and measures of effectiveness. It was proposed that the purpose of the previous JDL DF levels would be preserved by these new data fusion and resource management levels.

This section discusses some instances of research work that discuss PA/PE methodologies for data fusion processes, as well as issues on data/information fusion and resource management (subjects of management include signals/signatures, individual resources, coordinated resources, goals/mission objectives, system engineering and operational configuration) [26].

2.5.1 Performance Assessment/Evaluation Methodologies

In an earlier work, Oxenham et al. [250] introduced a rule-based expert system to perform high-level data fusion for human decision support. They described an updated version of the inference engine for that system in [250]. Measures of information were contrived for the assessment of system performance. These measures of information were combined to estimate the improvement or degradation in the information provided by the system output.

Rawat et al. [267] proposed a methodology in which the performance evaluation process for a data fusion-based multiple target tracking system was treated as an entirely new fusion process. The methodology was implemented to study the effect of track-truth association strategies on the performance metrics. The results showed that the selection of the track-truth association strategy should be done with reference to the scenario characteristics, the "mission" goals and the performance metrics to be evaluated. Llinas et al. [207] extended the work done in [267]. They introduced a detailed approach to the test planning and data analysis phases of PE based on formalized methods and associated analysis techniques. They also carried out various analyses and an additional case study associated with their suggested ideas for formalized planning and analysis

Type of data fusion system	Approach/Technique	Reference
General	Measures of input scenario complexity and output quality	[235]
Human decision support	Rule-based expert system	[250]
Multiple target tracking	Optimization-based hierarchical PE system, Statistical Design of Experiments (DOE), Analysis of Variance (ANOVA)	[207, 267]
Multi-sensor data fusion	Distributed fusion track-to-truth association, distributed fusion track-to-track association	[81]

Table 2.2: Performance assessment/evaluation for data fusion systems.

for PE.

Data fusion process designs being developed to tackle increasingly complex applications need to be assessed by technically viable and affordable performance evaluation methodology. Dastidar et al. [81] proposed a PE methodology for distributed Level 1 multi-sensor data fusion, with focus on applications that epitomized tactical aircraft systems. The interdependencies and the consistency measures between distributed fusion measures of performance were analyzed. The recommended PE methodology was implemented in a case study that involved track picture consistency across multiple platforms.

Ng et al. [235] presented a methodology to provide a well-rounded interpretation of data fusion system performance. The approach was based on metrics proposed to measure input scenario complexity and output quality. The metrics were combined to derive an assessment index to determine system performance. Experimental test results obtained for several scenarios verified the plausibility of the proposed evaluation tool.

Table 2.2 gives a summary of some approaches to performance assessment/evaluation for data fusion systems.

2.5.2 Data Fusion/Information Fusion and Resource Management

Multi-sensor Management (MSM) deals with the control of environment perception activities by the management or coordination of multiple sensor resource usage. It is an emerging research area and has become increasingly important in the research and development of modern multi-sensor systems for both military and civilian applications. Xiong and Svensson [334] provided a review of MSM in relation to multi-sensor information fusion. The work done included description of the role of MSM in the larger context, generalization of main problems from existing application needs and discussion on problem solving methodologies. In addition, many useful related works were cited. Some works that tackled issues on sensor management (SM) in detail are discussed in the remainder of this subsection.

Blackman and Popoli [22, Chapter 15] discussed principles and techniques for sensor management. The main issues of interest were: the necessity to include SM in the design of a modern sensor tracking system, the understanding of the aspects of sensor operation that required management and the figures of merit (metrics for the overall performance of an entire sensor tracking system) to be optimized by that management, as well as the approaches to accomplish SM.

Ng and Ng [231] studied the roles of sensor management, the motivation to use SM and presented a framework for a generic SM. Ng [230, Chapter 9] discussed classification and roles of SM and carried out simulation studies to demonstrate roles of SM as a controller.

Popp et al. [259] proposed an algorithm for dynamic sensor resource management (SRM) of an airborne multimode (ground moving target indicator and high range resolution) sensor for tracking and classification of ground moving targets. The SRM problem was cast within an optimization framework with multi-objective criteria. Simulation results for a ground target scenario demonstrated the inherent trade-offs between the radar modes and the multi-objective SRM criteria.

A stochastic dynamic programming (SDP) based approach to solving sensor resource management problems was described by Washburn et al. [323]. The SRM problem was formulated as a stochastic scheduling problem and approximate solutions based on the Gittins index rule were developed.

More recently, Johansson et al. [155] proposed the utilization of stochastic dynamic programming methods and algorithms for sensor resource allocation and management, an important part of future network-based defence (NBD). The more general method of reinforcement learning, of which SDP could be considered a special case, was also discussed. The authors reviewed some existing applications of SDP, as well as suggested the importance of SDP and reinforcement learning in building higher-level optimization and planning systems of future NBD systems.

Based on Shannon's entropy, Fassinut-Mombot and Choquel [99] described a novel probabilistic fusion methodology, named *Entropy Fusion Model*. The proposed fusion approach optimized the choice of measurements provided by information sources, which led to improved performance of the information fusion system. They developed the *Entropy Adaptative Aggregation* algorithm, an iterative method which used heuristic techniques to facilitate the implementation of the entropy fusion model. The performance and the robustness of the entropy adaptative aggregation algorithm were illustrated by experimental results obtained from an application to mobile robotics.

Xun et al. [335] presented an approach to receiver scheduling based on the control theory metaphor of software development. Experimental results showed that their control based sensor manager had significant advantages over a scheduler without feedback in terms of overload and the quality of service metric, the two metrics proposed for providing feedback on the quality of scheduler performance.

A Hierarchical Collective Agent Network was proposed by Zhu et al. [345] to address the problem of providing support for dynamic data fusion and management of a set of networked distributive sensors.

For optimal sensor resource allocation, Hill and Chang [53, 135] previously developed a model to provide hierarchical target valuation based on both Level 1 and Level 2 information fusion, using a joint kinematic and classification Markov chain model. They improved the model by modifying the existing Markov state transition models that allowed parameterization and approximate characterization of ground truth scenarios. Simulation results provided validation of the proposed approach. The authors also suggested extending the model to provide a higher fidelity approximation, as well as to account for Level 3 information.

Gonsalves and Rinkus [114] developed an intelligent fusion and asset management system to enhance tactical situation awareness and to provide useful information support for command and control personnel. The proposed system architecture comprised distinct modules for low-level fusion management, generation of probabilistic hypotheses for high-level situational state descriptors, as well as conversion of informational requirements and state information into asset resource requests. A battlefield scenario of defence against an adversary over a duration of one day was used to assess the feasibility of the system.

Molina et al. [224] proposed a fuzzy management scheme for evaluating multisensor tasks priority in defence surveillance applications. This approach allowed the integration of high-level information with conventional numeric representations in the decision process. The validity of the fuzzy reasoning approach was verified by the effectiveness of the fuzzy management scheme in managing environmental situations (in a similar way to that done by the experienced human operators).

The need to incorporate intelligent resource planner and manager capabilities within individual missile defence sensors was addressed by Burgess and Levins [45]. An intelligent sensor resource planning function based on evolutionary computing techniques (in particular, a genetic algorithm), was proposed. Simulation results demonstrated the feasibility of the proposed approach.

Dambreville and Le Cadre [71] reported the development of a general algebraic framework for multi-modal resource management of complex systems for detection/tracking of moving targets, under spatial and temporal constraints. The proposed framework could be adapted for various practical sensor management problems. Test results obtained from the implementation of the model demonstrated the efficiency of the system for managing solvable resource allocation problems and solving corresponding optimization problems.

Kreucher et al. [179] adopted an active sensing approach to manage agile sensors for multiple target tracking applications. The proposed method of sensor management was significantly more efficient than periodic scanning in the simulation scenarios considered.

To address the problem of sensor scheduling for detection and tracking of smart moving ground targets from an airborne sensor, Kreucher et al. [178] developed an algorithm with a reinforcement learning approach. Experimental results showed that the reinforcement learning approach was effective for performing quick target detection.

Liggins II and Chong [198] discussed the use of distributed fusion on multiple platforms for surveillance, as a means of SRM. Radar sensors with complementary capabilities were used on different types of platforms. Multiple fusion nodes were used to process sensor data locally. Locally processed information was subsequently communicated among the fusion nodes to achieve improvement over the individual local results.

Dhillon et al. [86] proposed an optimization framework for sensor resource management in distributed sensor networks. Their aim was to minimize the number of sensors deployed and the amount of sensed data reported, subject to the constraints of providing sufficient grid coverage of the sensor field and uncertainty associated with sensor detections. A (two or three-dimensional) grid representation of the sensor field was used. A sensor placement algorithm was developed to address coverage optimization for the sensor field, as well as to model preferential coverage of grid points (based on relative measures of security and tactical importance). Case studies were presented to discuss the superiority of the proposed approach over random sensor placement.

Chin et al. [63] discussed in detail an approach to the problem and challenge of managing multi-asset, heterogeneous sensor arrays in extended operating conditions. Results of experiments that verified the effectiveness of the proposed approach were also presented.

Research work on sensor resource management of naval multi-platform cooperative engagement was reported by Chen et al. [60]. An intelligent multi-agent based SRM structure was proposed. The aim was to achieve information superiority and subsequently, decision superiority, through effective management of information in the multi-platform cooperative engagement.

Komorniczak et al. [173] dealt with the problem of Multi Function Radar (MFR) resource management, which comprised target ranking and task scheduling. The focus was on the data fusion approach to the target ranking. Fuzzy and neural network systems were used to perform threat assessment and a comparison was made between them.

Zwaga and Driessen [346] earlier formulated the problem of finding an efficient MFR parameter control for single target tracking as a constrained minimization problem. The numerical solution for the proposed method determined the optimal MFR parameter control. In [346], they discussed the application of constraints on prediction accuracy for the proposed method. Simulation results showed that specific prediction accuracy could be maintained when using the proposed method. In addition, when compared with a typical conventional MFR parameter control, significantly lower amount of radar resources was needed.

Mårtenson and Svenson [214] introduced a general approach to sensor resource allocation. Their focus was on the problem of evaluating proposed sensor allocation schemes to provide optimized input and hence improve output, for a given fusion system. Formulation of the method was done in terms of random sets and equivalence classes of multi-target paths. The method was implemented in several test scenarios and its efficacy was demonstrated by the results obtained. They extended their research to develop a simulation-based tool for resource allocation and mission planning in military operations [312]. They proposed to improve the result through the utilization of *mixed-initiative interaction*, a relatively new area which emphasized efficient human-machine collaboration. Implementation of the proposed tool was discussed and suggestions for future work were given.

High-level information is playing an increasingly important role in research on sensor management. There is concern about the appropriateness in using the term *Sensor Management* to encompass the functions on the information level. In view of the necessity of using intelligent agents to perceive the environment to take suitable actions,

Application domain	Approach/Technique	Reference
Mobile robotics	Shannon's entropy-based probabilistic fusion of multiple information sources	[99]
Multiple sensor network	Probabilistic sensor placement algorithm coverage optimization	[86]
	Feedback control theory-based architecture for receiver/sensor scheduling	[335]
	Hierarchically networked agent architecture	[345]
Tactical defence	Genetic algorithm	[45]
- C4ISR	Intelligent multi-agent based SRM structure	[60]
- Military mission planning	Bayesian belief networks	[114]
- NBD/NCW	Fuzzy logic	[114, 224]
	Stochastic dynamic programming	[155]
	Distributed fusion on multiple platforms	[198]
	Random sets and equivalence classes of multi-target paths	[214]
	Simulation-based tool and mixed-initiative interaction	[312]
Target tracking	Bayesian network-based hierarchical target valuation algorithm	[53, 135]
- Ground target tracking	Hierarchical dynamic optimal control methods	[63]
and classification	Algebraic framework	[71]
- MFR tracking	Fuzzy logic, neural network system	[173]
- Multiple target tracking	Reinforcement learning	[178]
- Target detection	Machine learning (active sensing)	[179]
	Optimization-based dynamic algorithm (utilizes Markov models, decision trees)	[259]
	Stochastic dynamic programming	[323]
	Quadratic programming (numerical solver for constrained minimization problem)	[346]

Table 2.3: Data/information fusion & resource management: problems and techniques.

Johansson and Xiong [156] proposed a generic concept of *Perception Management* (PM), without having to be particular about concrete sensor device details. The concept referred to controlling the data acquisition process from the external world to enhance the perception outcomes. Two different possible interrelations between sensor management and perception management were considered and discussed: either SM is encompassed in PM or SM is separate from and independent of PM.

Table 2.3 provides a summary of some techniques applied for data fusion/information fusion and resource management in various problems.

2.6 Cognitive Refinement

Information representation and human-computer interaction are important for most data fusion systems. For example, it has been noted that the efficacy of the HCI had a significant influence on the overall performance and effectiveness of a data fusion

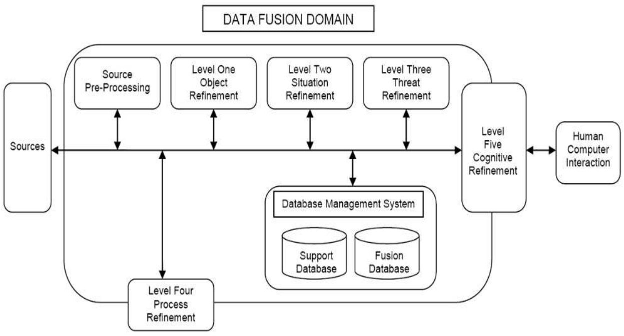

Figure 2.16: Augmented JDL DF model [123].

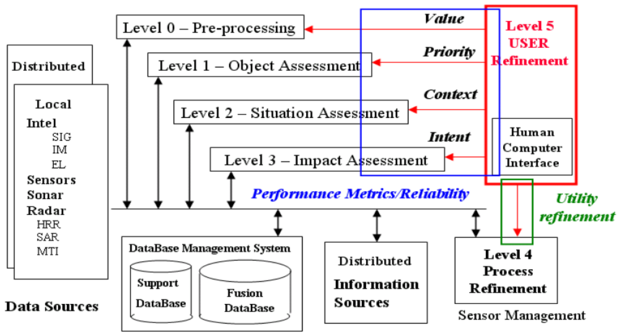

Figure 2.17: JDL-User model [28].

system [321]. On the other hand, the Object-Centered Information Fusion model [166] (see Section 2.1.4.1) took into consideration the role of a human for decision making.

The concept of Level 5 (*Cognitive Refinement*) processing in the original JDL DF model was introduced by Hall et al. [125] to account for functions associated with human-computer interaction explicitly. It involved the development of functions to support a human user in a collaborative human-computer environment. The categories of functions associated with Level 5 processing included [123]: HCI utilities, dialogue and transaction management and cognitive aids. Figure 2.16 shows the resultant augmented JDL DF model proposed. More discussion on various issues of cognitive refinement and human-computer interaction can be found in [123, Chapter 9].

In an independent work, Blasch and Plano [28] introduced Level 5 (*User* (or *Human*) *Refinement*, an element of Knowledge Management) with the purpose of supporting cognitive workload, trust, attention and situation awareness. In addition, the JDL-User model (shown in Figure 2.17) was proposed to extend the JDL DF model (the version in [304]) via the incorporation of the suggested Level 5. They explored further issues related to their concept of Level 5 (User Refinement) in [24, 25, 29–31].

More related research has been done recently. Hall et al. [124] discussed the development of a set of tools to support *whole-brain* information analysis (combines visually-oriented analysis of images with language-based analysis of text and related information). Nilsson and Ziemke [239] suggested adopting a distributed cognition perspective to complement existing approaches to understanding and modelling information fusion.

2.7 Applications

Since the introduction of data and information fusion techniques to the research community in the 1970s, the scope of application areas for data and information fusion has widened significantly. Some of the applications are discussed in the following subsections. Table 2.4 shows a summary of the techniques applied to the problems discussed.

2.7.1 Strategic/Tactical Defence

Data and information fusion was first used in military defence research related problems. After several decades of development, DIF techniques are now being developed and applied in diverse non-military research areas as well. Nevertheless, military defence research remains a very prominent application area for DIF [38,47]. Here, some research works from various defence applications are summarized.

Liggins II et al. [66,197] developed distributed architectures to support relevant fusion technologies such as multi-source fusion and sensor resource management. The technologies were applied to problems in defence and drug interdiction.

Gad and Farooq [105] discussed various data fusion architectures for maritime surveillance and developed a system that interacted with the data fusion processes at different information levels. This proposed data fusion architecture was shown to perform well when employed to support the maritime surveillance for a typical maritime tactical scenario.

Aldinger and Kao [3] discussed the challenges faced in undersea warfare and some research work done on developing data fusion technology and other techniques to enhance the capabilities of the undersea warfare community.

Ahlberg et al. [2] developed a concept demonstrator, the Information Fusion Demonstrator 2003 (IFD03), to demonstrate information fusion methodology expected to be suitable for a future network-based defence command, control, communications, computers, intelligence, surveillance and reconnaissance (C4ISR) system. The focus of IFD03 was on real-time intelligence processing in a tactical level ground warfare scenario. The architecture, methodology and user interface of the software system were described. The system was applied to a concrete scenario and related fusion results were discussed.

Introne et al. [149] developed a novel application that employed a two-level fusion architecture to address the problem of biosurveillance[2]. Feasibility of the approach was demonstrated via simulated outbreak events on a simulation platform.

2.7.2 Computer/Information Security

In the present age, where the use of information technology is ubiquitous, computer and information security issues are of great importance to both system administrators and general users. Information system issues such as intrusion detection in distributed communication and computer networks are receiving increasing amount of attention. Dasarathy [79] presented a general overview on research work done on intrusion detection.

Stein et al. [299] presented an outline on emerging concepts that were expected to guide future operations of joint military operations, as well as explained the achievement of information superiority via the use of network-centric computing. Experimental tests showed the effect of employing information superiority on the approach to fighting battles.

Browne [44] proposed that new approaches to command, control, communications, computers and intelligence (C4I) defensive architecture be developed to defend against multi-mode attacks, which were enemy strategies using clever combinations of conventional and non-conventional warfare. Criticism was made on some popular existing C4I defence technologies that were considered to be vulnerable against multi-mode attacks. A speculative discussion was presented on new C4I defence technologies and policy issues regarding information superiority that were believed to be inadequately addressed in existing literature.

A model based on multiple behaviour information fusion was developed for quantitative evaluation of network security threat by Chen et al. [62]. The proposed method was used for tests in a real network environment and was shown to be a reasonable and feasible tool for its system administrators.

Nicol [237] gave a discussion on using simulation to evaluate computer security in areas such as impact assessment (determine how security measures affect system and application performance) and emulation (combine real and virtual worlds to study the

[2]Biosurveillance: detection of attacks with unknown bioagents, also known as *syndromic surveillance*.

interaction between malware and systems, and probe for new system weaknesses).

2.7.3 Crisis/Disaster Management

In the event of a natural catastrophe or otherwise, there exists a large quantity of crucial data to be dealt with within a very short period of time immediately after the disaster [221]. It is essential to develop efficient data and information fusion tools for effective situation assessment and impact prediction in dynamic post-disaster scenarios, which in turn would be useful for decision making.

In view of the growing threats posed by potential use of chemical and biological agents in the military battlefield, Llinas et al. [206] addressed issues and challenges related to the development of technologies for effective combat against these weapons of mass destruction, in both military and civilian applications. Effective execution of battle management functions depends very much on high-quality information input. The authors asserted that it was very likely that the high-quality information demands of Nuclear, Chemical, Biological and Radiological (NCBR) battle management functions could be met by many existing information fusion techniques. In addition, it was possible for transition of advanced information fusion technologies from conventional warfare settings to NCBR-specific mission applications.

Llinas [204] described the overall strategic approach (engineering methodology) to a multi-year research program which addressed issues in information fusion to support crisis centre decision makers dealing with post-event situations. Both natural and man-made disasters were considered, with emphasis placed on post-earthquake and post-chemical attack scenarios respectively. The focus was on fusion capabilities at Levels 2 and 3 (higher-level information fusion). Examples of specific research components and subsequent research plans for the program were also discussed.

Little and Rogova [199] worked on the design of a general methodology for situation assessment to support crisis management. The proposed approach utilized understanding the combination of both formal and domain-specific construction methodologies and also described a general taxonomy of relationships, one which could encapsulate many of the complexities associated with catastrophic events.

A disaster monitoring interface for an earthquake simulation was proposed by Mandiak et al. [213]. The visualization tool was an integrated graphical user interface framework that enabled a user to easily comprehend the trend of a situation, by providing as much information (obtained via the integration of multidimensional graphic displays) as possible to him.

Rogova et al. [277] addressed the problem of situation assessment to support casualty mitigation operations in the response phase that immediately followed an earthquake. The proposed methodology was based on the cognitive work analysis and ontological analysis of a specific emergency management domain, developed within the framework of a formal ontology.

2.7.4 Fault Diagnosis

The main issues of concern when applying information fusion to fault diagnosis are the acquisition of reliable information about potential faults by incorporating multiple sensors, as well as the derivation of fused decisions based on data from the multiple sensors. It is necessary to develop fusion mechanisms that minimize conflicts among the sensors, as well as imprecision and uncertainty in the sensor data.

Based on Dempster-Shafer (D-S) evidence theory, a multi-sensor implementation of an engine diagnostic system was introduced by Basir and Yuan [14]. The formulation of the engine diagnostic problem in the context of the evidence theory was explained. Novel ways were introduced to enhance the effectiveness of mass functions in modelling and in evidence combination. Rational diagnosis decision making rules were proposed and the entropy of evidence was introduced to facilitate information fusion performance evaluation. Experimental results demonstrated the effectiveness of the proposed approach in resolving decision conflicts and in improving the accuracy of fault diagnosis via multi-sensor information fusion.

Fan and Zuo [97] introduced a Dempster-Shafer evidence theory based method with the capability of increasing accuracy of decision making through multi-source information fusion. In the proposed approach, fuzzy set theory, weight of evidence and conflict resolution were introduced to address the issues of evidence sufficiency, evidence importance, and conflicting evidence in the practical application of D-S evidence theory. Test example results validated feasibility of the proposed method, as well as its improvement over the conventional D-S evidence theory in performing fault diagnosis through fusing multi-source information. In the sequel [98], successful application of the improved D-S evidence theory to machinery fault diagnosis was reported. Experimental results showed that the proposed method could enhance diagnostic accuracy and autonomy, in comparison with conventional diagnostic methods.

2.7.5 Biomedical Applications/Informatics

Biomedical applications/informatics generally involves voluminous data from multiple heterogeneous sources. In most circumstances, the amount of useful knowledge that can be acquired from an individual data source is limited. Information derived from multi-source data fusion is often of better quality than that obtained from the available sources separately.

Bellot et al. [17] proposed a generic approach to fuse data in dynamical systems. A notion of qualified gain was defined to help determine the usefulness of a data fusion process developed. The method was applied to a problem of monitoring kidney disease patients who underwent dialysis at home. All the data sources and relations among them were determined. A dynamic Bayesian network based model was used to fuse the data in order to provide daily diagnosis on the hydration state of the patients. Efficiency of the proposed approach was reflected by the experimental results obtained.

Ganta et al. [106] described data exploration and analysis of heterogeneous biomedical informatics data sets using an online data warehouse. Experimental results obtained from applying information fusion techniques to multiple prostate cancer data sets demonstrated the feasibility of the proposed system.

Zhang et al. [343] presented a new approach to explore the cause of human longevity based on comprehensive medical data. Expert knowledge was applied to a longevity model through artificial intelligence techniques. Firstly, fuzzy logic was used in preprocessing biomedical data. Then multiple classifier network and decision level data fusion were applied to improve the modelling accuracy. Simulation test results showed that the proposed model was able to identify individuals who belong to longevity group with high accuracy.

Muller et al. [226] developed a modular data fusion system with Dempster-Shafer framework. An architecture of fusion was built from this system by chaining two types of elementary modules. The first type of modules were used for symbolic interpretation of numerical reports from sensors, while the second type were used for the combination of these symbolic data to obtain relevant synthetical information for diagnosis. The data fused were generated by tagged Magnetic Resonance Imaging[3] and Positron Emission

[3] Magnetic Resonance Imaging (MRI): an imaging technique based on the principles of Nuclear Magnetic Resonance, a spectroscopic technique used by scientists to elucidate chemical structure and molecular dynamics. MRI is used primarily in medical settings to produce high quality images of the inside of the human body.

Tomography[4]. D-S theory was applied to model the uncertainty of the data and the rules of decision. The fusion architecture was applied to the assessment of left ventricular myocardial viability[5]. To obtain geometrical information on the potential lesions, diagnosis results obtained from the data of a patient were displayed on polar maps.

A fuzzy logic based data fusion system for detection of life threatening patient states in cardiac care units was proposed by Kannathal et al. [95, 160]. Heterogeneous electrophysiological and haemodynamic data were fused and analyzed. In addition, a parameter named *patient deterioration index* was proposed to evaluate the severity of the cardiac abnormality. Test results obtained showed that the proposed approach could give highly accurate clinical diagnosis in monitoring the patients.

2.7.6 Environment

Human activities and environmental modifications can influence the ecosystem in multiple ways. The impact can be local, regional or even global. It is necessary to develop efficient systems to monitor and control activities that produce effects on the environment.

Hubert-Moy et al. [143] applied Dempster-Shafer's theory of evidence to support spatio-temporal monitoring and projections of land use and land cover changes. Data from spatial and temporal sources were fused to obtain spatial prediction of the location of winter bare fields for the following season on a watershed located in an intensive agricultural region. A highly accurate prediction on the presence of bare soils was achieved over the entire area of interest. The spatial distribution of misrepresented fields provided a good indicator for identification of change factors.

Heiden et al. [130] proposed a methodology to facilitate derivation of quantitative parameters for advanced evaluation of urban biotopes[6], an essential task in ecological urban planning. The proposed approach involved the analysis of airborne hyperspectral data and automated identification of urban surface cover types based on their material-specific spectral reflectance characteristics. The results were then integrated with vector-

[4]Positron Emission Tomography (PET): a highly specialized imaging technique that uses short-lived radioactive substances to produce three-dimensional colored images of those substances functioning within the body. These images are called PET scans and the technique is termed PET scanning.

[5]A *ventricle* is a heart chamber which collects blood from an *atrium* (another heart chamber that is smaller than a ventricle) and pumps it out of the heart. A *myocardium* is a muscular tissue of the heart. *Ventricular myocardial viability* is the potential for improvement of dysfunction in a ventricular myocardium after a surgical procedure for the provision of a new, additional, or augmented blood supply.

[6]Urban biotope: an area with uniform environment occupied by a unified urban community.

based urban biotope mapping, an existing database. Finally, the required quantitative parameters were derived from the resultant database. Spatial and statistical analyses showed that using quantitative parameters to complement the predominately descriptive information contained in urban biotope mapping yielded improved evaluation of urban biotopes.

Two data-driven tools, Support Vector Machines (SVM)[7] and Relevance Vector Machines (RVM)[8], were successfully applied to perform reliable soil moisture estimation by Khalil et al. [162]. The effectiveness and efficiency of the proposed models in soil moisture prediction were evaluated with the use of weather information. The performance and generalization capabilities of the two machines were also compared. SVM and RVM could be utilized in industries such as large scale water management to attain high-level inference via information, feature and decision level fusion processes.

In order to improve management of irrigation systems, good quality of spatial and temporal data on evapotranspiration (ETa), the combination of soil evaporation and plant transpiration, was essential. However, it was not easy to attain good quality for remote sensing ETa data. Chemin and Honda [57] reported an investigation on the use of genetic algorithms in assimilating parameters of an agrohydrological[9] model. The aim of the research was to find optimized parameters that would enable the model to obtain simulated ETa output that converged to observed remote sensing ETa data. The proposed methodology involved the fusion of observed remote sensing data of high spatial resolution, as well as those of low spatial resolution.

2.7.7 Industrial Applications

In the recent years, many industrial applications that utilize information fusion techniques for problem-solving have emerged [76]. Some instances of research work from prominent areas are reviewed below.

Qiu [262] presented the development of an effective data link between manufacturing and office planning to facilitate the deployment of an integrated plant-wide information system. The information-centric data fusion framework was proposed to help integrate

[7]Support vector machine: a constructive machine learning procedure based on statistical learning theory. It can be used to learn a variety of representations, such as neural nets, splines, and so on.

[8]Relevance vector machine: a machine learning technique based on Bayesian theory that has an identical functional form to the support vector machine.

[9]Agrohydrological: of or to do with *agrohydrology*, a research area that deals with climate, soil, and water and how these natural resources are managed in sustainable plant production.

all levels of data, with the aim of achieving synchronization and timely delivery of necessary information, in the information system. Details on the usefulness and practicality of the proposed model in the realization of a desired plant-wide real time information system were described.

A multi-layered fusion architecture and implementation for classifiers with binary and continuous outputs were described by Goebel and Yan [110]. The fusion scheme was structured into three major components which were partitioned into layers. The classifier outputs were transformed into a single continuous domain through logical tasks performed within the layers. The modular design of the fusion architecture allowed relatively easy addition/removal of modules, as well as the re-use of the core fusion engine for other domains. The proposed fusion framework was applied to a system monitoring environment of industrial equipment. The test results obtained were compared to those achieved by a a baseline approach. An improvement in performance over the latter was shown.

Roussel et al. [280] proposed a Bayesian inference based fusion method to combine the outputs of various sensors. The mathematical theory concerning the Bayesian approach was discussed and the method was applied to the problem of white grapes variety classification. The classification results verified the effectiveness of the proposed method in grape variety discrimination, an important task for manufacturers in the wine industry who need to determine accurately the origins and/or varieties of the grapes used for production.

Majidi and Moshiri [212] presented a computer vision system for classification of fruits. Estimation of the volume of a fruit was carried out by training a neural network with simple features of profile images of the fruit. Inspection of fruit surface defects was based on fusion of side images of the whole area of the fruit. A set of basic colour parameters of the fruit surface was then extracted and the fruit was classified via high level fusion of these visual features. Test results showed that the proposed method had acceptable performance in regard to the execution time required.

Ong and Ibañez-Guzmán [246] reviewed multi-sensor management for sensor fusion with respect to the guidance of unmanned vehicles. An information-oriented concept of perception management was introduced for multi-sensor systems. An outline of the concept of a design framework for sensor perception system was also given.

De Vin et al. [83, 84] reported how information fusion research could benefit manufacturing applications. One particular area of interest was virtual manufacturing. An

IF framework involving modelling and simulation was proposed for decision support in manufacturing. Relevant fused information regarding the past, present and future of the manufacturing system were extracted for future use. Interaction of the IF process with active databases (capable of propagating abnormal conditions or events to decision level), sensors and the simulation model was described. In [83], they also discussed some analogies between manufacturing and defence tasks, as well as aspects in which the manufacturing sector could benefit from defence research.

2.8 Summary

This chapter provides a survey on some process models that have been developed for data and information fusion. A review on some existing research work related to high-level information fusion, which is gaining interest in the recent years after much focus has been placed on low-level information fusion research, is also presented. Active research and development work on high-level information fusion is ongoing among the DIF community.

The variety of application areas which apply DIF techniques has increased tremendously since they were first applied in defence research in the 1970s. The scope of applications is still expanding fast, both in the military arena and civilian sectors (including commercial and industrial applications). Some examples of concepts and contexts with great potential for exploration include:

- network-centric warfare/operations and network-based defence [108, 248, 249, 281, 336],
- interoperability of joint and coalition military forces [88, 248, 249, 255, 328],
- information warfare [56, 91, 176, 183, 336],
- electronic and physical anomaly/intrusion detection [36, 46, 70, 79, 118, 126, 209, 225, 275, 306, 317, 318, 322, 338, 340],
- adversarial intent inference [16, 59, 100, 102, 107, 109, 154, 177, 232–234, 254, 275, 281, 290, 293, 303, 311, 313, 314, 338],
- biomedical applications/informatics and bioinformatics [51, 78, 123, 245, 284],
- human-computer interaction/human-machine interface [23, 80, 123, 125, 239],

Application domain	Approach/Technique	Reference
Strategic/tactical defence		
- Drug interdiction	Multiple platform distributed fusion	[66, 197]
- Maritime surveillance	Hybrid fusion: interaction with DF processes at different information levels	[105]
- Undersea warfare	Network-centric theatre undersea warfare architecture	[3]
- NBD C4ISR	IF methodology based on Dempster-Shafer clustering and template matching, particle filtering and finite set statistics	[2]
- Biosurveillance	Two-level fusion process based on information retrieval and dynamic Bayesian networks	[149]
Computer/information security evaluation	Multiple behaviour IF based on Markov models and Dempster-Shafer evidential reasoning	[62]
	Modelling and simulation for impact assessment, emulation, cyberattack exercises and training scenarios, and risk analysis/assessment based on known vulnerabilities exploits, attack capabilities and system configuration	[237]
Post-disaster management		
- Decision making	Engineering methodology utilizing Bayesian networks, Dempster-Shafer theory, fuzzy logic, neural networks	[204]
- Dynamic SA	Ontology meta-model	[199]
- DF visualization	Integrated GUI framework	[213]
- Casualty mitigation operations	Cognitive work analysis, ontological analysis	[277]
Engine/machinery fault diagnosis	Dempster-Shafer evidence theory-based multi-source IF	[14, 97, 98]
Biomedical Applications		
- Patient monitoring	Dynamic Bayesian network	[17]
- Data exploration/analysis	Multidimensional analysis, self-organizing map clustering algorithm	[106]
	Fuzzy logic, multiple classifier network, decision level DF	[343]
- Medical/clinical diagnosis	Dempster-Shafer framework	[226]
	Fuzzy logic	[95, 160]
Environment		
- Land monitoring and projection	Dempster-Shafer theory of evidence	[143]
- Ecological evaluation of urban biotopes	Spatial and statistical analyses of airborne hyperspectral data	[130]
- Soil moisture estimation	Support vector machines, relevance vector machines	[162]
- Irrigation system management	Genetic algorithm, agrohydrological model	[57]
Industrial applications		
- Information system deployment	Document object model for DF and aggregation	[262]
- System monitoring	Hierarchical, multi-layered fusion architecture	[110]
- Agricultural product quality control	Bayesian inference	[280]
	Neural network training	[212]
- Unmanned vehicle guidance	Information-oriented perception management	[246]
- Decision support in manufacturing	IF framework with modelling, resource simulation and active databases incorporated	[83, 84]

Table 2.4: Problems and techniques in various application areas.

- ontology-based approaches to high-level information fusion [40, 49, 54, 55, 148, 167, 189, 200–202, 221, 223, 228, 230, 241, 274, 297, 298, 341],
- resource management [4, 10, 18, 26, 41, 51, 90, 131–134, 138, 210, 227, 230, 266, 289, 295, 310, 313, 326, 327],
- image analysis/processing [50, 128, 252, 325].

With rapid advancement in various technologies and accessibility to vast data and information sources, complex information fusion problems are very likely to arise in many applications that involve far more concepts and contexts than the few listed above. It is becoming increasingly necessary to explore the possibility of expanding the base of diverse disciplines (including theories and techniques) upon which existing tools have been built. A lot more research is needed and can be done to develop novel useful tools (including theories, algorithms and architectures) for solving high-level information fusion problems. In addition, efficiency and effectiveness in this multidisciplinary field of research are likely to be enhanced if collaborative relationships can be established/strengthened among the various research groups [8, 203].

Chapter 3

Target Tracking

3.1 Introduction

In both the military and civilian sectors, increasingly high level of emphasis is placed on safety and security. This has led to widespread use and sophistication of systems for functions such as surveillance, guidance and obstacle avoidance. Consequently, much interest has been generated in the development of algorithms for target tracking, an essential component of these systems. An extensive survey on available target tracking systems and techniques is presented in [22].

Target tracking problems can be modelled by dynamic systems. For linear Gaussian problems, the Kalman filter (KF) [230] can be applied to obtain optimal solutions. The Kalman filter is easy to implement and has been popular since it was introduced in the 1960s. However, in practical applications, nonlinearity and/or non-Gaussianity often exist in target tracking problems. Nonlinear filtering techniques are required for modelling such problems.

For non-manœuvring target tracking, an extended Kalman filter (EKF) or an unscented Kalman filter (UKF) [9] is usually implemented to provide Gaussian approximation to the posterior probability density function (pdf) in the state space. The former uses the first-order Taylor series expansion of the nonlinear system equations that describe the given problem. The latter uses multiple deterministically chosen points in the state space to approximate the state distribution. Another group of popular techniques is the class of sequential Monte Carlo (MC) methods, also known as the particle filters (PFs) [9]. These methods do not have the limitation imposed by the Gaussian assumption required for EKFs and UKFs. Hence, they can be applied to problems with arbitrary nonlinearities or distributions. Random samples (or particles) are generated for the direct estimation of the posterior pdf. There is much research interest on sequential MC methods, especially over the past decade. Doucet et al. [87] and Ristic et al. [276] have detailed discussions on theoretical and practical aspects of these methods.

It is generally difficult to use a single model to represent the motion of a manœuvring target, as the manœuvres are often abrupt deviations from preceding motion. A comprehensive survey with detailed discussions on various issues pertaining to manœuvring target tracking is provided by the series of papers [191–196].

At present, multiple model based approaches are often used for manœuvring target tracking [68, 191, 230, 342]. The models in these methods run in parallel and describe different aspects of the target behaviour. In particular, the Interacting Multiple Model (IMM) algorithm [191, 222, 230] is widely accepted as one of the most cost-effective dynamic multiple model methods. It has been shown to achieve high performance with relatively low complexity. A complete cycle of the IMM filtering process consists of four essential operations, namely, mixing/interaction, filtering, mode probability update and combination. However, the mixing/interaction step yields a Gaussian mixture of posterior pdfs. The IMM algorithm is sub-optimal in the sense that it approximates this non-Gaussian element by a single Gaussian, which often results in serious errors.

In the recent years, researchers have developed techniques for solving nonlinear target tracking problems by combining multiple model based approaches (to account for mode switching) and particle filter variants (to account for nonlinear and/or non-Gaussian characteristics of the posed problem). A few instances are given below.

In [32], a particle filter with a switching/interaction step of the same form as that in the IMM algorithm was developed for stochastic hybrid systems. A regularized particle filter (RPF) was adopted in every model of the IMM algorithm in [35]. Instead of resampling, a Gaussian sum pdf was computed to approximate the conditional posterior pdf for the state in each mode. This method involved high computational complexity and additional approximations. In an improved approach [89], direct sampling from each mode conditional posterior pdf (a weighted sum of distributions) was implemented, in place of approximation with Gaussian mixtures. An unscented particle filter (UPF) and a generic/standard particle filter (SPF) were applied in each model of the IMM algorithm presented in [85] and [337] respectively. The multirate interacting multiple model particle filter developed in [139,140] placed emphasis on computation savings. The sample subset (particles) of each mode was updated at a different rate. An extended Kalman particle filter (EKPF) was used in each model. In [119], an interacting multiple models particle filter algorithm was proposed to deal with multiple noise corrupted measurements in nonlinear systems.

In this chapter, we implement an IMM algorithm for two problems on target tracking. The first problem involves the tracking of three-dimensional (3D) target motion with manœuvres. With the aim of covering various types of target behaviour, the following three models are used:

1. constant velocity (CV) model [22] - for uniform (non-manœuvring) motion;
2. constant acceleration (CA) model [22] - for manœuvre response;
3. 3D turning rate (3DTR) model [22, 324] - an extension of a coordinated turn (CT) model, which handles circular turn motion in a 2D horizontal plane, to a model for dealing with CT manœuvres in the 3D space.

The second problem is concerned with localization and tracking in a horizontal plane with the use of a Time Difference of Arrival (TDOA) system [61, 120]. As in the first problem, three models are used: a CV model, a CA model and a CT model.

Various combinations of nonlinear filters are used for the models [101]. Existing variants usually use an extended Kalman filter, an unscented Kalman filter or a particle filter in every model. The proposed approach uses a particle filter in the coordinated turn model, with extended Kalman filters and/or unscented Kalman filters in the remaining models. Simulation results show that new variants that use computationally economical particle filters in the coordinated turn models perform well for our test problems on manœuvring target tracking.

Next, we study a problem on modelling the prices of financial option contracts [240, 320]. The filtering algorithms discussed in this chapter are implemented for the tracking of financial option prices. A real data set is used in the numerical tests. The test results obtained are analyzed to assess the performance of the filters.

The structure of this chapter is as follows. Section 3.2 presents the formulation of the dynamical system model for manœuvring target tracking. Section 3.3 describes the IMM algorithm and the nonlinear filters to be used for the models. Section 3.4 contains details of the simulation tests conducted. Analysis and comparison of the numerical results obtained are also reported. Section 3.5 is focussed on the problem on pricing financial options. Section 3.6 discusses filter performance for the problems on manœuvring target tracking and modelling the prices of financial options. Section 3.7 summarizes this chapter.

3.2 Problem Formulation

A target tracking problem requires sequential estimation of the state of a dynamic system, based on a set of noisy measurements made on the system.

Consider a state-space model of a manœuvring target tracking system formulated as follows. Let k be the time index assigned to a continuous-time instant t_k. At time step k, X_k is an $n_x \times 1$ target state vector, Z_k is an $n_z \times 1$ measurement vector, e_k is an $n_e \times 1$ input information vector, w_k is an $n_w \times 1$ process noise vector and v_k is an $n_v \times 1$ (additive) measurement noise vector. Let $f(\cdot)$ be the system transition function, $g(\cdot)$ be the process noise input function and $h(\cdot)$ be the measurement function, with $f(\cdot)$ and $h(\cdot)$ possibly nonlinear. The system is represented by the process equation

$$X_k = f(X_{k-1}, t_{k-1}) + g(X_{k-1}, t_{k-1})w_{k-1}, \quad k \in \mathbb{N}, \tag{3.1}$$

and the measurement/observation equation

$$Z_k = h(X_k, e_k, t_k) + v_k, \quad k \in \mathbb{N}. \tag{3.2}$$

The states are assumed to follow a first-order Markov process and the observations are assumed to be independent given the states. Equation 3.1 defines the probabilistic model of the state evolution (transition prior), $p(X_k|X_{k-1})$, while Equation 3.2 defines the likelihood function of the current measurement given the current state, $p(Z_k|X_k)$. Let the initial pdf, $p(X_0|Z_0) := p(X_0)$, of the state vector (also known as the prior) be assumed to be available, with Z_0 being the set of no measurements. The complete solution to the state estimation problem is the true posterior pdf, $p(X_{0:k}|Z_{1:k})$, with $X_{0:k} := \{X_0, \ldots, X_k\}$ and $Z_{1:k} := \{Z_1, \ldots, Z_k\}$.

The objective is to estimate recursively in time $p(X_{0:k}|Z_{1:k})$. At an arbitrary time step k, by Bayes' Theorem [87],

$$p(X_{0:k}|Z_{1:k}) = \frac{p(Z_{1:k}|X_{0:k})p(X_{0:k})}{\int p(Z_{1:k}|X_{0:k})p(X_{0:k})dX_{0:k}}.$$

A recursive formula for $p(X_{0:k}|Z_{1:k})$ is

$$p(X_{0:k+1}|Z_{1:k+1}) = p(X_{0:k}|Z_{1:k})\frac{p(Z_{k+1}|X_{k+1})p(X_{k+1}|X_k)}{p(Z_{k+1}|Z_{1:k})}. \tag{3.3}$$

In order to obtain filtered estimates X_k based on the sequence of measurements $Z_{1:k}$ up to time step k, the marginal posterior pdf (filtering density/distribution) of the state, $p(X_k|Z_{1:k})$, needs to be computed recursively. This can be done via a recursive process of two stages, namely, prediction and update.

At time step k, assume the availability of the previous posterior pdf $p(X_{k-1}|Z_{1:k-1})$. The (dynamic) prior pdf (prediction density) of the state can be obtained at the prediction stage as [116, 276]

$$p(X_k|Z_{1:k-1}) = \int p(X_k|X_{k-1})p(X_{k-1}|Z_{1:k-1})dX_{k-1}. \tag{3.4}$$

When the measurement Z_k is received, the marginal posterior pdf is computed at the update stage as [116, 276]

$$p(X_k|Z_{1:k}) = \frac{p(Z_k|X_k)p(X_k|Z_{1:k-1})}{p(Z_k|Z_{1:k-1})}, \tag{3.5}$$

where

$$p(Z_k|Z_{1:k-1}) = \int p(Z_k|X_k)p(X_k|Z_{1:k-1})dX_k$$

is a normalizing constant, which depends on $p(Z_k|X_k)$ and the known statistics of v_k.

For most nonlinear problems, the posterior pdf cannot be determined analytically. Hence, tractable filtering methods are required to obtain approximate solutions for such problems.

3.3 Filtering Algorithms

The nonlinear filters to be used for the models in the IMM algorithm in this thesis are briefly discussed in this section. Details on the filters can be found in the respective cited references.

3.3.1 Extended Kalman Filters

Extended Kalman filters [13, 276] are commonly used for solving nonlinear target tracking problems. An EKF is a minimum mean square error estimator that uses first-order Taylor series expansions to approximate nonlinear system functions in Equations 3.1 and 3.2. It provides a Gaussian approximation to the marginal posterior pdf of the system state through its conditional mean and covariance.

At time step k, let $\hat{x}(k-1|k-1)$ and $\hat{P}(k-1|k-1)$ denote the estimates of the mean and covariance of X_{k-1}, given $Z_{1:k-1}$, respectively. Let the expression $\mathcal{N}(y; \bar{y}, \Sigma)$ denote the pdf (or density) of a multivariate Gaussian (normal) random variable y, where

$$\bar{y} = E(y) \quad \text{and} \quad \Sigma = E[(y-\bar{y})(y-\bar{y})^T]$$

are the mean and the covariance matrix of y respectively. The following recursive relationships are assumed to hold:

$$p(X_{k-1}|Z_{1:k-1}) \approx \mathcal{N}(X_{k-1}; \hat{x}(k-1|k-1), \hat{P}(k-1|k-1)), \tag{3.6}$$

$$p(X_k|Z_{1:k-1}) \approx \mathcal{N}(X_k; \hat{x}(k|k-1), \hat{P}(k|k-1)), \tag{3.7}$$

$$p(X_k|Z_{1:k}) \approx \mathcal{N}(X_k; \hat{x}(k|k), \hat{P}(k|k)). \tag{3.8}$$

The implementation of an extended Kalman filter[1] is stated below, where $F(k)$ and $G(k)$ are Jacobians of the process equation, $H(k)$ is the Jacobian of the measurement equation, $Q(k)$ is the covariance of the process noise, $R(k)$ is the covariance of the measurement noise, and $K(k)$ is the filter gain.

1. Prediction:
   $\hat{x}(k|k-1) = f(\hat{x}(k-1|k-1))$,
   $\hat{P}(k|k-1) = F(k)\hat{P}(k-1|k-1)F(k)^T + G(k)Q(k)G(k)^T$.

2. Update:
   $S(k) = H(k)\hat{P}(k|k-1)H(k)^T + R(k)$,
   $K(k) = \hat{P}(k|k-1)H(k)^T S(k)^{-1}$,
   $\hat{z}(k|k-1) = h(\hat{x}(k|k-1))$,
   $\tilde{z}(k) = z(k) - \hat{z}(k|k-1)$,
   $\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\tilde{z}(k)$,
   $\hat{P}(k|k) = \hat{P}(k|k-1) - K(k)S(k)K(k)^T$.

For problems with mild nonlinearity, an extended Kalman filter provides satisfactory results with much efficiency. However, when problems are highly nonlinear and the effects of the higher-order terms of the Taylor series expansions are not negligible, approximation results obtained with an EKF are prone to errors. In such situations, the filter is likely to perform badly or even diverge. Two examples of approaches which aim to alleviate the effects of errors due to linearization in the EKFs are briefly discussed below [13, 276].

Higher-order extended Kalman filters take into consideration first-order, as well as higher-order, terms in the Taylor series expansions. However, they are not widely used because of increased complexity and implementation costs. The additional complexity and costs are caused by the need for the computation of Jacobian and Hessian matrices in the higher-order terms of the Taylor series expansions. In cases when these matrices are complicated in derivation, it could be very computationally expensive to compute them at every step of the algorithms. There is also no guarantee of improved results.

The iterated extended Kalman filter (IEKF) differs from the EKF in the update stage. The updated state is computed as a maximum a posteriori (MAP) estimate

[1] As the class of extended Kalman filters require analytical evaluation of the Jacobians, these filters cannot be applied in cases when the model functions are discontinuous.

instead of as an approximate conditional mean (a linear combination of the prediction and the innovation). This is equivalent to relinearization of the measurement equation around the *updated* state instead of the *predicted* state. An iterative process such as Newton-Raphson algorithm is carried out to obtain the required MAP estimate, with the use of the current measurement vector. The maximum number of iterations is decided beforehand or based on a convergence criterion. The IEKF has been reported to perform very well only in the rare practical situation in which the measurement model fully observes the state.

3.3.2 Unscented Kalman Filters

An unscented Kalman filter [158, 320] is also a minimum mean square error estimator. Unlike the case of extended Kalman filters, explicit calculation of Jacobians or Hessians is not required for the implementation of a UKF. For an arbitrary nonlinear problem, it uses a minimal set of deterministically chosen sample points (or "sigma points") to obtain a Gaussian approximation to the marginal posterior pdf of the system state. The selected sample points capture the actual mean and covariance of the Gaussian density completely. After propagation through the nonlinear system, the posterior mean and covariance of the state computed from the transformed samples are accurate up to the second order (third order for a Gaussian prior) of the Taylor series expansions.

At time step k, let $\hat{x}(k-1|k-1)$ and $\hat{P}(k-1|k-1)$ denote the estimates of the mean and covariance of X_{k-1}, given $Z_{1:k-1}$ respectively. The following gives the implementation of an unscented Kalman filter.

1. Compute $2n_x+1$ sigma points, $\{\zeta^{(i)}(k-1|k-1)\}_{i=0}^{2n_x}$, and the associated weights, $\{W_i\}_{i=0}^{2n_x}$, with $\sum_{i=0}^{2n_x} W_i = 1$. Let κ be a scaling parameter. For an $n_x \times n_x$ matrix M, let $(\sqrt{M})_i$ denote the i-th column or row of the matrix square root of M obtained via Cholesky decomposition [111], $i = 1, \ldots, n_x$.

$$
\begin{aligned}
&\zeta^{(i)}(k-1|k-1) \\
&= \begin{cases}
\hat{x}(k-1|k-1), & i = 0, \\
\hat{x}(k-1|k-1) + \left(\sqrt{(n_x+\kappa)\hat{P}(k-1|k-1)}\right)_i, & i = 1, \ldots, n_x, \\
\hat{x}(k-1|k-1) - \left(\sqrt{(n_x+\kappa)\hat{P}(k-1|k-1)}\right)_{i-n_x}, & i = n_x+1, \ldots, 2n_x;
\end{cases}
\end{aligned}
$$

$$
W_i = \begin{cases}
\dfrac{\kappa}{(n_x+\kappa)}, & i = 0, \\
\dfrac{1}{2(n_x+\kappa)}, & i = 1, \ldots, n_x, \\
\dfrac{1}{2(n_x+\kappa)}, & i = n_x+1, \ldots, 2n_x.
\end{cases}
$$

2. Propagation.

   Propagate the sigma points through the dynamic system:

   $\zeta^{(i)}(k|k-1) = f(\zeta^{(i)}(k-1|k-1)), \quad i = 0, \ldots, 2n_x.$

   Compute the predicted mean and covariance of the state:

   $\hat{x}(k|k-1) = \sum_{i=0}^{2n_x} W_i \zeta^{(i)}(k|k-1),$

   $\hat{P}(k|k-1) = G(k)Q(k)G(k)^T + \sum_{i=0}^{2n_x} W_i[\zeta^{(i)}(k|k-1) - \hat{x}(k|k-1)][\zeta^{(i)}(k|k-1) - \hat{x}(k|k-1)]^T.$

3. Update.

   Compute the measurement sigma points and predicted measurement:

   $\xi^{(i)}(k|k-1) = h(\zeta^{(i)}(k|k-1)), \quad i = 0, \ldots, 2n_x,$

   $\hat{z}(k|k-1) = \sum_{i=0}^{2n_x} W_i \xi^{(i)}(k|k-1).$

   Compute the updated mean and covariance of the state:

   $P_{xz} = \sum_{i=0}^{2n_x} W_i[\zeta^{(i)}(k|k-1) - \hat{x}(k|k-1)][\xi^{(i)}(k|k-1) - \hat{z}(k|k-1)]^T,$

   $P_{zz} = \sum_{i=0}^{2n_x} W_i[\xi^{(i)}(k|k-1) - \hat{z}(k|k-1)][\xi^{(i)}(k|k-1) - \hat{z}(k|k-1)]^T,$

   $S(k) = P_{zz} + R(k),$

   $K(k) = P_{xz} S(k)^{-1},$

   $\tilde{z}(k) = z(k) - \hat{z}(k|k-1),$

   $\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\tilde{z}(k),$

   $\hat{P}(k|k) = \hat{P}(k|k-1) - K(k)S(k)K(k)^T.$

The above form of implementation is valid for cases with additive noise and $n_x + \kappa$ is constant [320, 332, 333]. The general form of the algorithm can be found in [158, 320].

UKFs have been reported to outperform EKFs in many problems at no additional computational costs (the two groups of filters have almost the same computational complexity, each being $O(n_x^3)$ [13, 82]. However, UKFs also have the limitation that they cannot be applied to general non-Gaussian problems.

3.3.3 Particle Filters

Particle filters are sequential Monte Carlo methods that use finite (usually large) set of samples/particles to directly approximate a required probability density. Basic ideas on PFs were first introduced in Physics and Statistics in the 1950s [127, 278]. Research on these techniques was relatively limited during the 1960s and the 1970s. This was probably due to the lack of powerful machines for computation in those days.

Most particle filters developed over the past years are based on sequential importance sampling (SIS) [9]. The introduction of the idea of resampling [116] and advances in computing facilities made great contribution to the practicality of PFs. Research on PFs has been active over the past decade. With the aid of powerful computers, much improvement on PFs has been attained and numerous application areas have evolved for these techniques.

3.3.3.1 Monte Carlo Methods

Monte Carlo integration methods form the basis for sequential MC methods [87,276,320]. Consider a multidimensional integral

$$I(f) = \int f(x)p(x)dx, \quad x \in \mathbb{R}^{n_x}, \tag{3.9}$$

such that $p(\cdot)$ is interpreted as a pdf, that is,

$$p \geq 0 \quad \text{and} \quad \int p(x)dx = 1.$$

Assume that a set $\{x^{(i)}\}_{i=1}^{N_s}$ of $N_s \gg 1$ independent and identically distributed (i.i.d.) samples/particles can be simulated according to $p(\cdot)$. Then the integral $I(f)$ can be approximated by the sample mean

$$I_{N_s}(f) = \frac{1}{N_s}\sum_{i=1}^{N_s} f(x^{(i)}),$$

which is an unbiased estimate. By the Strong Law of Large Numbers (see Theorem A.2), $I_{N_s}(f) \xrightarrow{a.s.} I(f)$ as $N_s \to \infty$. If the variance of $f(\cdot)$,

$$\sigma_f^2 = \int (f(x) - I(f))^2 p(x)dx < \infty,$$

then the Central Limit Theorem (see Theorem A.1) holds and

$$\sqrt{N_s}(I_{N_s}(f) - I(f)) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \sigma_f^2),$$

where the expression $\mathcal{N}(\mu, \sigma^2)$ denotes the pdf (or density) of a Gaussian (normal) random variable with mean μ and variance σ^2 (standard deviation σ).

As the error of the MC estimate, $I_{N_s}(f) - I(f)$, is of order $O(N_s^{-1/2})$, the rate of convergence of the estimate is independent of the dimension of the integrand n_x.

It is generally not possible to generate samples from $p(\cdot)$ directly. One can use the *importance sampling* method to surmount this problem. In place of $p(\cdot)$, use a *proposal distribution* (also known as *importance sampling distribution* or *importance density function*) $q(\cdot)$, such that the support of $q(\cdot)$ contains the support of $p(\cdot)$. The MC integral in Equation 3.9 can be rewritten as

$$I(f) = \frac{\int f(x)\frac{p(x)}{q(x)}q(x)dx}{\int \frac{p(x)}{q(x)}q(x)dx}, \tag{3.10}$$

provided that $p(x)/q(x)$ has an upper bound for all $x \in \mathbb{R}^{n_x}$.

To compute an MC estimate of $I(f)$, first generate $N_s \gg 1$ i.i.d. samples $\{x^{(i)}\}_{i=1}^{N_s}$ according to q. Then estimate $I(f)$ by a weighted sum [87, 276]

$$\hat{I}_{N_s}(f) = \frac{\frac{1}{N_s}\sum_{i=1}^{N_s} f(x^{(i)})\tilde{w}(x^{(i)})}{\frac{1}{N_s}\sum_{i=1}^{N_s} \tilde{w}(x^{(i)})} = \sum_{i=1}^{N_s} f(x^{(i)})w(x^{(i)}),$$

where

$$\tilde{w}(x^{(i)}) = \frac{p(x^{(i)})}{q(x^{(i)})} \quad \text{and} \quad w(x^{(i)}) = \frac{\tilde{w}(x^{(i)})}{\sum_{j=1}^{N_s} \tilde{w}(x^{(j)})}, \quad i = 1, \ldots, N_s,$$

are the unnormalized and normalized importance weights respectively.

The importance sampling technique is applied in the Bayesian framework with $p(\cdot)$ being the posterior pdf.

3.3.3.2 Sequential Importance Sampling

By making importance sampling, a general Monte Carlo technique recursive, one obtains the sequential importance sampling algorithm. This simple and general MC method, forms the basis for most of particle filters that have been developed over the last few decades [87, 276, 320]. The required posterior pdf is represented by a set of random particles with associated weights. Posterior estimates are then computed based on these samples and their weights. As the sample size, N_s, becomes increasingly large, by the Strong Law of Large Numbers (see Theorem A.2), the approximation converges to the true posterior pdf [320].

It is generally not possible to draw samples from the posterior density directly. Hence, in practice, samples are drawn from a known proposal distribution, $q(X_k|X_{0:k-1}, Z_{1:k})$, instead. Then a corresponding importance weight is computed for each particle.

Let $\{X_{0:k}^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$ denote a random measure that characterizes $p(X_{0:k}|Z_{1:k})$, where $\{X_{0:k}^{(i)}\}_{i=1}^{N_s}$ is a set of support points with associated normalized weights $\{w_k^{(i)}\}_{i=1}^{N_s}$, that is, $\sum_{i=1}^{N_s} w_k^{(i)} = 1$.

The importance density from which the samples are drawn can be of the form

$$q(X_{0:k}|Z_{1:k}) = q(X_{0:k-1}|Z_{1:k-1})q(X_k|X_{0:k-1}, Z_{1:k}), \tag{3.11}$$

such that the previously simulated state $X_{0:k-1}$ are not modified during the computation of the posterior distribution at time step k. In addition, it is assumed that the current state does not depend on future observations. Based on the assumption that the state dynamics is a Markov process and that the observations are conditionally independent given the states,

$$p(X_{0:k}) = p(X_0)\prod_{j=1}^{k} p(X_j|X_{j-1}) \quad \text{and} \quad p(Z_{1:k}|X_{0:k}) = \prod_{j=1}^{k} p(Z_j|X_j). \tag{3.12}$$

By Equations 3.11 and 3.12, a recursive estimate for the associated unnormalized importance weight of each particle $X_k^{(i)}$, $i = 1, \ldots, N_s$, can be derived as [320]

$$\begin{aligned}
\tilde{w}_k^{(i)} &= \frac{p(Z_{1:k}|X_{0:k}^{(i)})p(X_{0:k}^{(i)})}{q(X_{0:k}^{(i)}|Z_{1:k})} \\
&= \frac{p(Z_{1:k}|X_{0:k}^{(i)})p(X_{0:k}^{(i)})}{q(X_{0:k-1}^{(i)}|Z_{1:k-1})q(X_k^{(i)}|X_{0:k-1}^{(i)}, Z_{1:k})} \\
&= \frac{\tilde{w}_{k-1}^{(i)}}{p(Z_{1:k-1}|X_{0:k-1}^{(i)})p(X_{0:k-1}^{(i)})} \frac{p(Z_{1:k}|X_{0:k}^{(i)})p(X_{0:k}^{(i)})}{q(X_k^{(i)}|X_{0:k-1}^{(i)}, Z_{1:k})} \\
&= \tilde{w}_{k-1}^{(i)} \frac{p(Z_k|X_k^{(i)})p(X_k^{(i)}|X_{k-1}^{(i)})}{q(X_k^{(i)}|X_{0:k-1}^{(i)}, Z_{1:k})},
\end{aligned} \tag{3.13}$$

with normalization

$$w_k^{(i)} = \tilde{w}_k^{(i)} \left[\sum_{j=1}^{N_s} \tilde{w}_k^{(j)}\right]^{-1}. \tag{3.14}$$

A discrete weighted approximation to $p(X_{0:k}|Z_{1:k})$ can subsequently be obtained as

$$\hat{p}(X_{0:k}|Z_{1:k}) = \sum_{i=1}^{N_s} w_k^{(i)} \delta(X_{0:k} - X_{0:k}^{(i)}), \tag{3.15}$$

where $\delta(\cdot)$ is the Dirac delta measure defined below.

Definition 3.1 *Dirac delta measure (or Dirac (impulse) delta function).*

The Dirac delta measure $\delta(\cdot)$ is defined by

$$\delta(x) = \begin{cases} \infty, & x = 0, \\ 0, & x \neq 0; \end{cases} \qquad \text{and} \qquad \int_{-\infty}^{\infty} \delta(x)dx = 1. \tag{3.16}$$

□

On sequential reception of each measurement, support points and their associated importance weight are recursively propagated via the SIS algorithm, as described by Algorithm 1.

Algorithm 1: Sequential Importance Sampling

Input: $\{X_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_s}$, Z_k

1. For $i = 1, \ldots, N_s$,

a. draw sample $X_k^{(i)}$ from the proposal distribution $q(X_k|X_{0:k-1}, Z_{1:k})$;

b. compute importance weight

$$\tilde{w}_k^{(i)} = \tilde{w}_{k-1}^{(i)} \frac{p(Z_k|X_k^{(i)})p(X_k^{(i)}|X_{k-1}^{(i)})}{q(X_k^{(i)}|X_{0:k-1}^{(i)}, Z_{1:k})};$$

End For.

2. Compute total weight: $t = \sum_{i=1}^{N_s} \tilde{w}_k^{(i)}$.

3. For $i = 1, \ldots, N_s$,

normalize the importance weight: $w_k^{(i)} = t^{-1}\tilde{w}_k^{(i)}$;

End For.

Output: $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$

The choice of importance density is a critical issue in the design of a particle filter. The optimal importance density,

$$q(X_k|X_{0:k-1}, Z_{1:k}) = p(X_k|X_{0:k-1}, Z_{1:k}), \tag{3.17}$$

minimizes the variance of the importance weights, conditional on $X_{0:k-1}$ and $Z_{1:k}$ [320]. It is usually not easy to find the optimal proposal distribution, so suboptimal alternatives are normally used in practice. A convenient choice of proposal distribution adopted by conventional particle filters is the transition prior,

$$q(X_k|X_{0:k-1}, Z_{1:k}) = p(X_k|X_{k-1}), \tag{3.18}$$

which is simple and easy to implement. Consequently, the unnormalized importance weights in Equation 3.13 can be simplified as

$$\tilde{w}_k^{(i)} = \tilde{w}_{k-1}^{(i)} p(Z_k|X_k^{(i)}), \quad i = 1, \ldots, N_s, \tag{3.19}$$

with normalization described by Equation 3.14.

In contrast with the general importance density $q(X_k|X_{0:k-1}, Z_{1:k})$, it is not possible to incorporate the current measurement data Z_k when using the transition prior $p(X_k|X_{k-1})$ as the importance density. This may cause serious deficiency in particle filters, especially when $p(X_k|X_{k-1})$ has a much broader distribution than the likelihood function $p(Z_k|X_k)$. Only a few particles generated by the transition prior will have significant importance weights, while the others can easily land in the regions of low likelihood and thus wasted.

One problem that often arises with the SIS algorithm is the *degeneracy phenomenon* in which all but very few of the importance weights become negligible over time. Equivalently, one of the particle importance weights approaches 1, while the remaining ones all approach 0. In this situation, much computational effort is used for the update of particles that have almost no contribution to $p(X_k|Z_{1:k})$.

An appropriate measure of degeneracy of an algorithm is the *effective sample size* defined as [9, 276]

$$N_{\text{eff}} = \frac{N_s}{1 + \text{Var}(W_k^{(i)})}, \tag{3.20}$$

where

$$W_k^{(i)} = \frac{p(X_k^{(i)}|Z_{1:k})}{q(X_k^{(i)}|X_{k-1}^{(i)}, Z_k)}, \quad i = 1, \ldots, N_s, \tag{3.21}$$

is referred to as the "true weight" and $\text{Var}(\cdot)$ denotes variance. It is not possible to evaluate N_{eff} exactly and it is usually estimated by

$$\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^{N_s} (w_k^{(i)})^2} \tag{3.22}$$

in practice, with $\{w_k^{(i)}\}_{i=1}^{N_s}$ being the normalized weights obtained using Equations 3.13 and 3.14. It is noted that $N_{\text{eff}} \leq N_s$. Severe degeneracy is indicated by small N_{eff}. One approach to overcome the problem of degeneracy is to use very large N_s, but it is very often impractical.

To alleviate the adverse effects of degeneracy of the SIS algorithm, resampling can be carried out to eliminate particles with low importance weights and multiply particles with high importance weights, with the number of particles N_s kept unchanged [116, 276, 320]. Resampling is to be carried out when degeneracy is significant, that is, when N_{eff} falls below some predetermined threshold N_T [9, 276, 283]. In this thesis, the threshold used is $N_T = 2N_s/3$.

Through resampling, a random measure $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$ is mapped into a new random measure $\{X_k^{(i)*}, w_k^{(i)*}\}_{i=1}^{N_s}$ [276, 320] with uniform weights $w_k^{(i)*} = 1/N_s$, $i = 1, \ldots, N_s$. The new set $\{X_k^{(i)*}\}_{i=1}^{N_s}$ is generated by resampling (with replacement) N_s times from an approximate discrete representation of $p(X_k|Z_{1:k})$,

$$\hat{p}(X_k|Z_{1:k}) = \sum_{i=1}^{N_s} w_k^{(i)} \delta(X_k - X_k^{(i)}), \tag{3.23}$$

with $\text{Prob}\{X_k^{(i)*} = X_k^{(j)}\} = w_k^{(j)}$. The resulting sample is an independent identically distributed sample from the discrete density in Equation 3.23, with uniform importance weights $w_k^{(i)*} = 1/N_s$, $i = 1, \ldots, N_s$. Resampling can be carried out as follows [276, 320].

The cumulative sum of normalized weights (CSW) of the random measure

$\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$ is constructed as

$$\text{CSW}(i) = \begin{cases} 0, & i = 0, \\ \sum_{j=1}^{i} w_k^{(j)}, & i = 1, \ldots, N_s. \end{cases}$$

A set of random samples $\{u_i\}_{i=1}^{N_s}$ is generated from the Uniform distribution on $[0,1]$, sorted in ascending order and compared with the CSW. For $i = 1, \ldots, N_s$, when the condition $u_i \in (\text{CSW}(j-1), \text{CSW}(j)]$ is satisfied by some integer $j = 1, \ldots, N_s$, the corresponding sample $X_k^{(j)}$ is accepted as a new sample: $X_k^{(i)*} = X_k^{(j)}$.

A variety of resampling schemes is available in the literature [9, 276, 320]. As the specific choice of resampling scheme does not affect the performance of the particle filter, systematic resampling is used in this thesis (see Algorithm 2). It is an efficient scheme that is simple to implement and minimizes the MC variation. Its computational complexity is $O(N_s)$.

Algorithm 2: Systematic Resampling

Input: $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$

1. Initialize the CSW: $c_1 = w_k^{(1)}$.
2. For $j = 2, \ldots, N_s$,
   construct CSW: $c_j = c_{j-1} + w_k^{(j)}$;
   End For.
3. Start at the bottom of the CSW: $j = 1$.
4. Draw a starting point, u_1, from $\mathcal{U}[0, N_s^{-1}]$, the Uniform distribution on $[0, N_s^{-1}]$.
5. For $i = 1, \ldots, N_s$,
   a. move along the CSW: $u_i = u_1 + (i-1)N_s^{-1}$;
   b. While $u_i > c_j$,
      $j = j + 1$;
      End While;
   c. assign sample: $X_k^{(i)*} = X_k^{(j)}$;
   d. assign weight: $w_k^{(i)*} = N_s^{-1}$;
   e. assign parent: $\eta^i = j$;

   End For.

Output: $\{X_k^{(i)*}, w_k^{(i)*}, \eta^i\}_{i=1}^{N_s}$

Resampling brings about practical problems such as those listed below [9, 276].

1. The opportunity to parallelize is limited because all particles must be combined.

2. Particles with high importance weights are statistically selected many times, which yields a resultant sample that will contain many repeated points in the state space, leading to a loss of diversity among the particles and a poor representation of the posterior pdf. This problem is known as *sample impoverishment* and is severe in cases of small process noise.
3. Due to the reduction in the diversity of the paths of the particles, any smoothed estimates based on these paths degenerate.

3.3.3.3 Generic/Standard Particle Filter

The sequential importance sampling filter and resampling techniques discussed above form the core composition of the standard particle filter, which is described by Algorithm 3 [9].

Algorithm 3: Generic/Standard Particle Filter

Input: $\{X_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_s}$, Z_k

1. Compute $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$ via SIS (Algorithm 1).
2. Calculate $\hat{N}_{\text{eff}} = \left[\sum_{i=1}^{N_s}(w_k^{(i)})^2\right]^{-1}$.
3. If $\hat{N}_{\text{eff}} < N_T$
   carry out resampling to generate equally weighted particles $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$;
   End If.
4. Compute the state estimate and the corresponding state error covariance:
   $\hat{X}_k = \sum_{i=1}^{N_s} w_k^{(i)} X_k^{(i)}$,
   $\hat{P}_k = \sum_{i=1}^{N_s} w_k^{(i)}[X_k^{(i)} - \hat{X}_k][X_k^{(i)} - \hat{X}_k]^T$.

Output: $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$, $\hat{X}_k$, $\hat{P}_k$

3.3.3.4 Auxiliary Particle Filter

The sampling importance resampling (SIR) filter [9, 116] is a technique for the implementation of a recursive Bayesian filter by Monte Carlo simulations [87, 276, 320]. It is derivable from the sequential importance sampling algorithm. The transition prior is used as the importance density and resampling is carried out at every time step.

The advantages of the SIR filter are the relative ease in evaluating the importance weights and sampling from the importance density (transition prior, which is independent of the observations). However, the filter has the potential of being inefficient and sensitive to outliers, due to the exploration of the state space without any knowledge of the observations. In addition, applying resampling at every time step can cause rapid loss in the diversity of the particles.

The auxiliary particle filter (APF) [9,256,276], also known as the auxiliary sampling importance resampling (ASIR) filter, was proposed as a variant of the standard SIR filter. The APF can be derived from the sequential importance sampling algorithm by utilizing an auxiliary variable, i, and the current measurement Z_k at time step k, to introduce an importance density $q(X_k, i|Z_{1:k})$. The importance density is used to draw samples $\{X_k^{(j)}, \eta^j\}_{j=1}^{N_s}$, where η^j is the index of the particle at time step $k-1$, $j = 1, \ldots, N_s$. Let $\mu_k^{(i)}$ be a point estimate that characterizes $p(X_k|X_{k-1}^{(i)})$, $i = 1, \ldots, N_s$. The importance density is defined such that

$$q(X_k, i|Z_{1:k}) \propto p(Z_k|\mu_k^{(i)})p(X_k|X_{k-1}^{(i)})w_{k-1}^{(i)}. \tag{3.24}$$

In this thesis, $\mu_k^{(i)}$ is a sample from $p(X_k|X_{k-1}^{(i)})$, $i = 1, \ldots, N_s$. The preceding proportionality, together with the factorization

$$q(X_k, i|Z_{1:k}) = q(i|Z_{1:k})q(X_k|i, Z_{1:k}), \tag{3.25}$$

and the definition

$$q(X_k|i, Z_{1:k}) := p(X_k|X_{k-1}^{(i)}), \tag{3.26}$$

yields

$$q(i|Z_{1:k}) \propto p(Z_k|\mu_k^{(i)})w_{k-1}^{(i)}. \tag{3.27}$$

For $j = 1, \ldots, N_s$, the sample $\{X_k^{(j)}, \eta^j\}$ is assigned an importance weight

$$\tilde{w}_k^{(j)} \propto \tilde{w}_{k-1}^{(\eta^j)} \frac{p(Z_k|X_k^{(j)})p(X_k^{(j)}|X_{k-1}^{(\eta^j)})}{q(X_k^{(j)}, \eta^j|Z_{1:k})} = \frac{p(Z_k|X_k^{(j)})}{p(Z_k|\mu_k^{(\eta^j)})}. \tag{3.28}$$

The points which the auxiliary particle filter generates from the sample at time step $k-1$ are highly likely to be close to the true state, conditioned on the current measurement. This provides the APF with an advantage over the SIR filter. The APF can be regarded as resampling at the previous time step, based on point estimates that characterize the transition prior. When process noise is small (respectively, large), $\{p(X_k|X_{k-1}^{(i)})\}_{i=1}^{N_s}$ is well (respectively, badly) characterized by $\{\mu_k^{(i)}\}_{i=1}^{N_s}$ and the APF is likely to perform better (respectively, worse) than the SIR filter. Algorithm 4 presents one cycle of the auxiliary particle filter.

Algorithm 4: Auxiliary Particle Filter

Input: $\{X_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_s}$, Z_k

1. For $i = 1, \ldots, N_s$,

   a. obtain $\mu_k^{(i)}$: draw sample $\mu_k^{(i)}$ from the proposal distribution $p(X_k|X_{k-1}^{(i)})$;

   b. compute importance weight

$$\tilde{w}_k^{(i)} = q(i|Z_{1:k}) \propto p(Z_k|\mu_k^{(i)})w_{k-1}^{(i)};$$

   End For.

2. Compute total weight: $t = \sum_{i=1}^{N_s} \tilde{w}_k^{(i)}$.

3. For $i = 1, \ldots, N_s$,

   normalize the importance weight: $w_k^{(i)} = t^{-1}\tilde{w}_k^{(i)}$;

   End For.

4. Generate $\{\eta^j\}_{j=1}^{N_s}$ via resampling from $\{\mu_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$.

5. For $j = 1, \ldots, N_s$,

   a. draw sample $X_k^{(j)}$ from the proposal distribution $p(X_k|X_{k-1}^{(\eta^j)})$;

   b. compute importance weight

$$\tilde{w}_k^{(j)} \propto \tilde{w}_{k-1}^{(\eta^j)} \frac{p(Z_k|X_k^{(j)})p(X_k^{(j)}|X_{k-1}^{(\eta^j)})}{q(X_k^{(j)}, \eta^j|Z_{1:k})} = \frac{p(Z_k|X_k^{(j)})}{p(Z_k|\mu_k^{(\eta^j)})};$$

   End For.

6. Compute total weight: $t = \sum_{i=1}^{N_s} \tilde{w}_k^{(i)}$.

7. For $i = 1, \ldots, N_s$,

   normalize the importance weight: $w_k^{(i)} = t^{-1}\tilde{w}_k^{(i)}$;

   End For.

8. Calculate $\hat{N}_{\text{eff}} = \left[\sum_{i=1}^{N_s}(w_k^{(i)})^2\right]^{-1}$.

9. If $\hat{N}_{\text{eff}} < N_T$

   carry out resampling to generate equally weighted particles $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$;

   End If.

10. Compute the state estimate and the corresponding state error covariance:

    $\hat{X}_k = \sum_{i=1}^{N_s} w_k^{(i)} X_k^{(i)}$,

    $\hat{P}_k = \sum_{i=1}^{N_s} w_k^{(i)}[X_k^{(i)} - \hat{X}_k][X_k^{(i)} - \hat{X}_k]^T$.

Output: $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$, $\hat{X}_k$, $\hat{P}_k$

3.3.3.5 Regularized Particle Filter

Regularized particle filters [9,276] were developed with the aim of providing a remedy for the problem of sample impoverishment. A regularization step is added when resampling is being conducted in the standard particle filter. The kernel method is used to obtain a regularized empirical representation of the marginal posterior pdf, $p(X_k|Z_{1:k})$, by

$$\hat{p}(X_k|Z_{1:k}) = \sum_{i=1}^{N_s} w_k^{(i)} K_\beta(X_k - X_k^{(i)}) \tag{3.29}$$

where

$$K_\beta(x) = \frac{1}{\beta^{n_x}} K\left(\frac{x}{\beta}\right) \tag{3.30}$$

is the rescaled kernel density $K(\cdot)$, $\beta > 0$ is the kernel bandwidth (a scalar parameter) and $w_k^{(i)}$, $i = 1, \ldots, N_s$, are normalized importance weights. The kernel density is a symmetric pdf such that

$$K \geq 0, \quad \int K(x)dx = 1, \quad \int xK(x)dx = 0, \quad \int \|x\|^2 K(x)dx < \infty.$$

The kernel density and kernel bandwidth are chosen such that the mean integrated square error (MISE) between $p(X_k|Z_{1:k})$ and $\hat{p}(X_k|Z_{1:k})$,

$$\text{MISE}(\hat{p}) = E\left[\int [\hat{p}(X_k|Z_{1:k}) - p(X_k|Z_{1:k})]^2 dX_k\right] \tag{3.31}$$

is minimized. In this thesis, the Gaussian kernel is used and the optimal choice for the bandwidth is

$$\beta_{\text{opt}} = AN_s^{-\frac{1}{n_x+4}} \quad \text{with} \quad A = [4/(n_x+2)]^{\frac{1}{n_x+4}}. \tag{3.32}$$

The proposal distribution used is the transition prior.

In practice, the RPF usually performs better than the sampling importance resampling filter in cases of severe sample impoverishment, such as when process noise is small. However, the RPF has a theoretical disadvantage that the samples generated are not guaranteed to have an asymptotic approximation to those from the marginal posterior. Algorithm 5 describes one cycle of the regularized particle filter.

Algorithm 5: Regularized Particle Filter

Input: $\{X_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N_s}$, Z_k

1. For $i = 1, \ldots, N_s$,
   a. draw sample $X_k^{(i)*}$ from the proposal distribution $p(X_k|X_{k-1}^{(i)})$;
   b. compute importance weight $\tilde{w}_k^{(i)} = p(Z_k|X_k^{(i)*})$;

   End For.
2. Compute total weight: $t = \sum_{i=1}^{N_s} \tilde{w}_k^{(i)}$.
3. For $i = 1, \ldots, N_s$,
   normalize the importance weight: $w_k^{(i)*} = t^{-1}\tilde{w}_k^{(i)}$;

   End For.

4. Calculate $\hat{N}_{\text{eff}} = \left[\sum_{i=1}^{N_s} (w_k^{(i)*})^2\right]^{-1}$.
5. If $\hat{N}_{\text{eff}} < N_T$
   a. calculate the empirical covariance matrix S_k for $\{X_k^{(i)*}, w_k^{(i)*}\}_{i=1}^{N_s}$;
   b. compute D_k such that $D_k D_k^T = S_k$;
   c. carry out resampling to generate equally weighted particles $\{\tilde{X}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$ from $\{X_k^{(i)*}, w_k^{(i)*}\}_{i=1}^{N_s}$;
   d. For $i = 1, \ldots, N_s$,
      i. draw sample $\Gamma^{(i)}$ from the Gaussian kernel;
      ii. compute $X_k^{(i)} = \tilde{X}_k^{(i)} + \beta_{\text{opt}} D_k \Gamma^{(i)}$;

      End For.

   End If.
6. Compute the state estimate and the corresponding state error covariance:
   $\hat{X}_k = \sum_{i=1}^{N_s} w_k^{(i)} X_k^{(i)}$,
   $\hat{P}_k = \sum_{i=1}^{N_s} w_k^{(i)} [X_k^{(i)} - \hat{X}_k][X_k^{(i)} - \hat{X}_k]^T$.

Output: $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$, $\hat{X}_k$, $\hat{P}_k$

The regularized particle filter differs from the standard particle filter in the regularization steps that are required during resampling. The empirical covariance matrix S_k is calculated before resampling and is a function of $\{X_k^{(i)*}, w_k^{(i)*}\}_{i=1}^{N_s}$.

3.3.3.6 Extended Kalman Particle Filter

As mentioned in Section 3.3.3.2, the proposal distribution used in the standard particle filter, the transition prior, does not take into consideration the current measurement data. Consequently, deficiency may arise in particle filters, especially when there is little overlap between the proposal distribution and the posterior pdf. In this situation, it is desirable to move the samples in the prior towards regions of high likelihood.

To avoid problems that may arise from using the transition prior as the proposal distribution, researchers have developed various linearization-based approaches which incorporate the current measurement to obtain a Gaussian approximation to the optimal proposal distribution. One example is the extended Kalman particle filter [320].

At time step k, on receiving the measurement Z_k, a separate extended Kalman filter is used to compute the mean $\hat{X}_k^{(i)}$ and the covariance $\hat{P}_k^{(i)}$ of a Gaussian proposal distribution,

$$q(X_k^{(i)} | X_{0:k-1}^{(i)}, Z_{1:k}) = \mathcal{N}(X_k^{(i)}; \hat{X}_k^{(i)}, \hat{P}_k^{(i)}), \tag{3.33}$$

for the i-th ($i = 1, \ldots, N_s$) particle in this framework.

An EKF usually propagates particles towards the likelihood function and a possibly better proposal distribution is likely to be generated, with more overlap between the proposal distribution and the posterior pdf being attained. However, this requires a Gaussian assumption on the form of the posterior. Linearization may also result in inaccuracies. Algorithm 6 shows a single cycle of the extended Kalman particle filter.

Algorithm 6: Extended Kalman Particle Filter

Input: $\{X_{k-1}^{(i)}, P_{k-1}^{(i)}\}_{i=1}^{N_s}$, Z_k

1. For $i = 1, \ldots, N_s$,

   a. run an EKF to generate updated mean $\hat{X}_k^{(i)}$ and covariance $\hat{P}_k^{(i)}$;

   b. draw sample $\tilde{X}_k^{(i)}$ from the proposal distribution $\mathcal{N}(X_k^{(i)}; \hat{X}_k^{(i)}, \hat{P}_k^{(i)})$;

   c. compute importance weight

   $$\tilde{w}_k^{(i)} = \tilde{w}_{k-1}^{(i)} \frac{p(Z_k|\tilde{X}_k^{(i)})p(\tilde{X}_k^{(i)}|X_{k-1}^{(i)})}{q(\tilde{X}_k^{(i)}|X_{0:k-1}^{(i)}, Z_{1:k})},$$

   where $q(\tilde{X}_k^{(i)}|X_{0:k-1}^{(i)}, Z_{1:k}) = \mathcal{N}(\tilde{X}_k^{(i)}; \hat{X}_k^{(i)}, \hat{P}_k^{(i)})$;

   End For.

2. Compute total weight: $t = \sum_{i=1}^{N_s} \tilde{w}_k^{(i)}$.

3. For $i = 1, \ldots, N_s$,
   normalize the importance weight: $w_k^{(i)} = t^{-1}\tilde{w}_k^{(i)}$;
   End For.

4. Calculate $\hat{N}_{\text{eff}} = \left[\sum_{i=1}^{N_s} (w_k^{(i)})^2\right]^{-1}$.

5. If $\hat{N}_{\text{eff}} < N_T$
   resample to get $\{X_k^{(i)}, P_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$, with the importance weights being equal;
   End If.

6. Compute the state estimate and the corresponding state error covariance:
   $\hat{X}_k = \sum_{i=1}^{N_s} w_k^{(i)} X_k^{(i)}$,
   $\hat{P}_k = \sum_{i=1}^{N_s} w_k^{(i)} \{P_k^{(i)} + [X_k^{(i)} - \hat{X}_k][X_k^{(i)} - \hat{X}_k]^T\}$.

Output: $\{X_k^{(i)}, P_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$, $\hat{X}_k$, $\hat{P}_k$

3.3.3.7 Unscented Particle Filter

For each particle in the framework of the extended Kalman particle filter, use an unscented Kalman filter instead of an extended Kalman filter for generating the proposal distribution. The resultant filter is known as the unscented particle filter [320].

3.3.3.8 Gaussian Particle Filter

The Gaussian particle filter (GPF) is an instance of several particle filtering techniques that do not require resampling [129, 174]. Like sequential importance sampling based particle filters, the GPF uses importance sampling to obtain particles. The GPF propagates only the mean and covariance of the posterior pdf. However, higher moments can be propagated due to the fact that all moments can be estimated using importance sampling.

The GPF usually outperforms conventional Gaussian filters such as extended Kalman filters and unscented Kalman filters, especially for solving problems with nontrivial nonlinearities. It also has lower computational complexity than particle filters which require resampling, a process that may be computationally expensive.

At time step k, let $\hat{x}(k-1|k-1)$ and $\hat{P}(k-1|k-1)$ denote the estimates of the mean and covariance of X_{k-1}, given $Z_{1:k-1}$, respectively. The GPF approximates the posterior pdf $p(X_{k-1}|Z_{1:k-1})$ by a single Gaussian distribution,

$$\hat{p}(X_{k-1}|Z_{1:k-1}) = \mathcal{N}(X_{k-1}; \hat{x}(k-1|k-1), \hat{P}(k-1|k-1)), \tag{3.34}$$

similar to Gaussian filters such as EKFs. The Gaussian particle filter is carried out as follows.

1. Draw samples from the Gaussian approximation to the previous posterior pdf:
   $x^{(i)}(k-1|k-1) \sim \hat{p}(X_{k-1}|Z_{1:k-1}), \quad i = 1, \ldots, N_s.$

2. Prediction.
   Propagate the samples through the dynamic system:
   $x^{(i)}(k|k-1) = f(x^{(i)}(k-1|k-1)), \quad i = 1, \ldots, N_s.$
   Compute the predicted mean and covariance of the state:
   $\hat{x}(k|k-1) = \sum_{i=1}^{N_s} w_{k-1}^{(i)} x^{(i)}(k|k-1),$
   $\hat{P}(k|k-1) = \sum_{i=1}^{N_s} w_{k-1}^{(i)} [x^{(i)}(k|k-1) - \hat{x}(k|k-1)][x^{(i)}(k|k-1) - \hat{x}(k|k-1)]^T.$
   Approximate the prior pdf $p(X_k|Z_{1:k-1})$ by the Gaussian distribution

   $$\hat{p}(X_k|Z_{1:k-1}) = \mathcal{N}(X_k; \hat{x}(k|k-1), \hat{P}(k|k-1)). \tag{3.35}$$

3. Update.
   Draw samples from the proposal distribution:
   $x^{(i)}(k|k) \sim q(X_k|Z_{1:k}), \quad i = 1, \ldots, N_s.$
   Compute the associated unnormalized importance weights

   $$\tilde{w}_k^{(i)} = \frac{p(Z_k|x^{(i)}(k|k))\hat{p}(x^{(i)}(k|k)|Z_{1:k-1})}{q(x^{(i)}(k|k)|Z_{1:k})}, \quad i = 1, \ldots, N_s, \tag{3.36}$$

and obtain the normalized weights $\{w_k^{(i)}\}_{i=1}^{N_s}$ via Equation 3.14.

Compute the updated mean and covariance of the state:

$\hat{x}(k|k) = \sum_{i=1}^{N_s} w_k^{(i)} x^{(i)}(k|k)$,

$\hat{P}(k|k) = \sum_{i=1}^{N_s} w_k^{(i)} [x^{(i)}(k|k) - \hat{x}(k|k)][x^{(i)}(k|k) - \hat{x}(k|k)]^T$.

When the proposal distribution is chosen as

$$q(X_k|Z_{1:k}) = \hat{p}(X_k|Z_{1:k-1}), \tag{3.37}$$

the unnormalized importance weights in Equation 3.36 become simplified as

$$\tilde{w}_k^{(i)} = p(Z_k|x^{(i)}(k|k)), \quad i = 1, \dots, N_s, \tag{3.38}$$

with the use of $X_k = x^{(i)}(k|k)$ in Equation 3.37. The normalization is described by Equation 3.14.

The Gaussian particle filter has also been used as building blocks for the Gaussian sum particle filter [175], which can be applied to models that cannot approximate the posterior pdf well with single Gaussians, as well as models with non-Gaussian noise.

3.3.4 The Interacting Multiple Model Algorithm

The IMM algorithm is built from several dynamic motion models that represent different target behavioural traits. The models can switch from one to another according to a set of transition probabilities governed by an underlying Markov chain. One complete cycle of the IMM algorithm comprises four parts, namely, an input mixer (interaction), a filter for each model, a model probability evaluator and an output mixer (combination).

The flow diagram of an IMM algorithm with r models is shown in Figure 3.1, where $M_j(k)$ denotes model j at time step k, $j = 1, \dots, r$. Here, $r = 3$. Table 3.1 shows the various combinations of filters considered for the respective models. An outline of the k-th cycle of a typical IMM algorithm variant implemented in this thesis is given below [101, 230, 324].

Step 1: Interaction

Mode-conditioned state estimates and state error covariances of the previous step are merged using mixing probabilities for the initialization of the current step. For $M_j(k)$, compute

- the initial state estimate,

$$\hat{x}_j^0(k-1|k-1) = \sum_{i=1}^{r} \hat{x}_i(k-1|k-1)\mu_{i|j}(k-1|k-1),$$

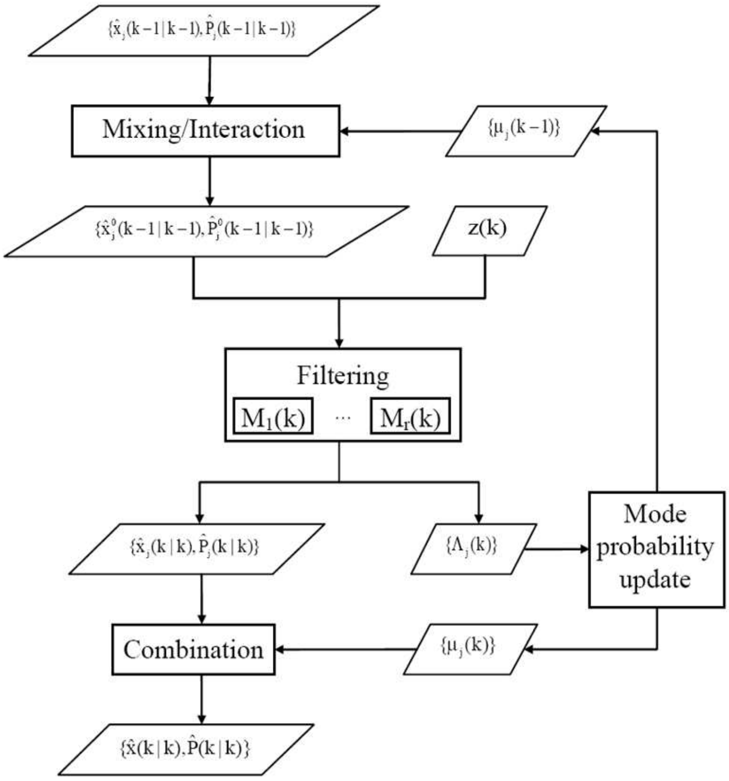

Figure 3.1: The IMM algorithm (r models).

- and the corresponding state error covariance,

$$\hat{P}_j^0(k-1|k-1) = \sum_{i=1}^{r} \mu_{i|j}(k-1|k-1)\big\{\hat{P}_i(k-1|k-1) + [\tilde{x}_{ij}(k-1)][\tilde{x}_{ij}(k-1)]^T\big\},$$

where

$\hat{x}_j(k-1|k-1)$ is the prior state estimate,

$\hat{P}_j(k-1|k-1)$ is the corresponding state error covariance,

$\mu_{i|j}(k-1|k-1) = \bar{c}_j^{-1} p_{ij}\mu_i(k-1)$ is mixing probability,

$\bar{c}_j = \sum_{i=1}^{r} p_{ij}\mu_i(k-1)$ is a normalizing constant,

$\tilde{x}_{ij}(k-1) = \hat{x}_i(k-1|k-1) - \hat{x}_j^0(k-1|k-1)$, and

p_{ij} is the assumed transition probability for switching from model i (at time step $k-1$) to model j (at time step k).

Step 2: Filtering

Determine the relevant filter from Table 3.1. Through the use of the initial state estimate and its corresponding state error covariance from Step 1, as well as an exogenous measurement data $z(k)$, model updates for $M_j(k)$ are performed by computing

- the state estimate $\hat{x}_j(k|k)$, and

- the corresponding state error covariance $\hat{P}_j(k|k)$.

Notation for IMM algorithm variant	Model 1 (M_1)	Model 2 (M_2)	Model 3 (M_3)
IEK	EKF	EKF	EKF
IUK	UKF	UKF	UKF
IEE	EKF	EKF	EKPF
IEG	EKF	EKF	GPF
IER	EKF	EKF	RPF
IES	EKF	EKF	SPF
IEU	EKF	EKF	UPF
IEA	EKF	EKF	APF
IUE	UKF	UKF	EKPF
IUG	UKF	UKF	GPF
IUR	UKF	UKF	RPF
IUS	UKF	UKF	SPF
IUU	UKF	UKF	UPF
IUA	UKF	UKF	APF
IEUE	EKF	UKF	EKPF
IEUG	EKF	UKF	GPF
IEUR	EKF	UKF	RPF
IEUS	EKF	UKF	SPF
IEUU	EKF	UKF	UPF
IEUA	EKF	UKF	APF

Table 3.1: Filters used for the models in the IMM algorithm variants.

The input set for each filter is described as follows.

Extended Kalman filter or unscented Kalman filter:

use $\hat{x}_j^0(k-1|k-1)$ and $\hat{P}_j^0(k-1|k-1)$ as $\hat{x}(k-1|k-1)$ and $\hat{P}(k-1|k-1)$ respectively for the implementation of EKF (Section 3.3.1) or UKF (Section 3.3.2).

Standard particle filter, auxiliary particle filter or regularized particle filter:

- draw a set $\{\hat{x}_j^{(i)}(k-1|k-1)\}_{i=1}^{N_s}$ of N_s samples from the a priori pdf approximated by $\mathcal{N}(\hat{x}_j^0(k-1|k-1), \hat{P}_j^0(k-1|k-1))$;
- use $\{\hat{x}_j^{(i)}(k-1|k-1)\}_{i=1}^{N_s}$ and the importance weights $\{w_j^{(i)}(k-1)\}_{i=1}^{N_s}$ at time step $k-1$ as inputs for the implementation of SPF (Section 3.3.3.3), APF (Section 3.3.3.4) or RPF (Section 3.3.3.5).

Extended Kalman particle filter or unscented particle filter:

- draw a set $\{\hat{x}_j^{(i)}(k-1|k-1)\}_{i=1}^{N_s}$ of N_s samples from the a priori pdf approximated by $\mathcal{N}(\hat{x}_j^0(k-1|k-1), \hat{P}_j^0(k-1|k-1))$;
- set the corresponding covariances $\{\hat{P}_j^{(i)}(k-1|k-1)\}_{i=1}^{N_s}$, with $\hat{P}_j^{(i)}(k-1|k-1) := \hat{P}_j^0(k-1|k-1)$, $i = 1, \ldots, N_s$;

- use $\{\hat{x}_j^{(i)}(k-1|k-1), \hat{P}_j^{(i)}(k-1|k-1)\}_{i=1}^{N_s}$ as inputs for the implementation of EKPF (Section 3.3.3.6) or UPF (Section 3.3.3.7).

Gaussian particle filter:

use $\hat{x}_j^0(k-1|k-1)$ and $\hat{P}_j^0(k-1|k-1)$ as $\hat{x}(k-1|k-1)$ and $\hat{P}(k-1|k-1)$ respectively for the implementation of GPF (Section 3.3.3.8).

Step 3: Mode probability update

For $M_j(k)$, with the use of filter residual $\tilde{z}_j(k)$, the corresponding filter residual covariance $\hat{S}_j(k)$ and the assumption of a Gaussian distribution, the likelihood function is computed as

$$\Lambda_j(k) = \frac{1}{\sqrt{\det(2\pi\hat{S}_j(k))}} \exp\{-0.5(\tilde{z}_j(k))^T(\hat{S}_j(k))^{-1}\tilde{z}_j(k)\},$$

where $\det(M)$ the determinant of the square matrix M. The mode probability of $M_j(k)$ is then updated as

$$\mu_j(k) = \frac{1}{c}\Lambda_j(k)\bar{c}_j,$$

where $c = \sum_{i=1}^{r} \Lambda_i(k)\bar{c}_i$.

Step 4: Combination

All the state estimates and their corresponding state error covariances output from the individual models are utilized for the computation of

- the combined state estimate,

$$\hat{x}(k|k) = \sum_{i=1}^{r} \mu_i(k)\hat{x}_i(k|k),$$

- and the corresponding state error covariance,

$$\hat{P}(k|k) = \sum_{i=1}^{r} \mu_i(k)\{\hat{P}_i(k|k) + [\hat{x}_i(k|k) - \hat{x}(k|k)][\hat{x}_i(k|k) - \hat{x}(k|k)]^T\}.$$

3.4 Simulation Tests and Results

We carry out simulation tests to evaluate of the IMM algorithm variants listed in Table 3.1. The following test problems are considered:

1. manœuvring target tracking in three-dimensional space, and
2. target tracking using a time difference of arrival system.

3.4.1 Manœuvring Target Tracking in Three-dimensional Space

With reference to Equations 3.1 and 3.2, consider a target tracking system represented by the process equation:

$$X_{k+1} = f(X_k, t_k, m_k) + g(X_k, t_k, m_k)w_k, \tag{3.39}$$

and the measurement/observation equation

$$Z_{k+1} = h(X_{k+1}, t_{k+1}, m_{k+1}) + v_{k+1}, \tag{3.40}$$

where $X_k = [x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k, \ddot{x}_k, \ddot{y}_k, \ddot{z}_k]^T$ is the state vector, and $Z_k = [x_k, y_k, z_k]^T$ is the measurement vector.

At time step k, the state vector X_k is in mode $m_k \in \{1, 2, 3\}$, with m_k being the modal state of the system. The process noise vector w_k is zero-mean multivariate Cauchy-distributed (multivariate t distribution, with one degree of freedom) with correlation matrix Q_j for model j, $j = 1, 2, 3$. Let I_3 and 0_3 denote the 3×3 identity and zero matrices respectively. The state evolutions within the different modes are given as follows.

Model 1 (M_1 - CV model):

$$f(X_k, t_k, 1) = \begin{bmatrix} I_3 & TI_3 & 0_3 \\ 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 \end{bmatrix} X_k, \quad g(X_k, t_k, 1) = \begin{bmatrix} 0.5T^2 I_3 \\ TI_3 \\ 0_3 \end{bmatrix}, \quad Q_1 = 5^2 I_3.$$

Model 2 (M_2 - CA model):

$$f(X_k, t_k, 2) = \begin{bmatrix} I_3 & TI_3 & 0.5T^2 I_3 \\ 0_3 & I_3 & TI_3 \\ 0_3 & 0_3 & I_3 \end{bmatrix} X_k, \quad g(X_k, t_k, 2) = \begin{bmatrix} 0.5T^2 I_3 \\ TI_3 \\ I_3 \end{bmatrix}, \quad Q_2 = 20^2 I_3.$$

Model 3 (M_3 - 3DTR model):

$$f(X_k, t_k, 3) = \begin{bmatrix} I_3 & \frac{\sin(\omega_k T)}{\omega_k} I_3 & \frac{(1-\cos(\omega_k T))}{\omega_k^2} I_3 \\ 0_3 & \cos(\omega_k T) I_3 & \frac{\sin(\omega_k T)}{\omega_k} I_3 \\ 0_3 & -\omega_k \sin(\omega_k T) I_3 & \cos(\omega_k T) I_3 \end{bmatrix} X_k, \quad g(X_k, t_k, 3) = \begin{bmatrix} 0.5T^2 I_3 \\ TI_3 \\ I_3 \end{bmatrix},$$

$Q_3 = 10^2 I_3$, where $\omega_k = \dfrac{\|[\ddot{x}_k, \ddot{y}_k, \ddot{z}_k]\|_2}{\|[\dot{x}_k, \dot{y}_k, \dot{z}_k]\|_2}$ is the turning rate.

The measurement function is mode-independent, given by

$$h(X_k, t_k, m_k) = \begin{bmatrix} I_3 & 0_3 & 0_3 \end{bmatrix} X_k.$$

The measurement noise vector v_k is zero-mean Gaussian with covariance $R = 10^5 I_3$.

The transition probability matrix and the initial mode probability are

$$\begin{bmatrix} 0.96 & 0.02 & 0.02 \\ 0.02 & 0.96 & 0.02 \\ 0.02 & 0.02 & 0.96 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0.96 & 0.02 & 0.02 \end{bmatrix}$$

respectively. The sampling interval (time interval between successive scans) is $T = 1$ (second). For each particle filter used, the number of particles is $N_s = 300$. The total number of independent simulation runs is $C = 100$.

For this test problem, IEK (IMM algorithm using an extended Kalman filter in every model) is used as the basis case for comparison of the filtering algorithms implemented.

3.4.1.1 Scenarios

Most of the existing research literature on 3D target tracking focus on target manœuvres in a horizontal plane (at constant altitude). In general, target manœuvres occur in the 3D space instead of just in a horizontal plane. We consider simulated target trajectories from both of the above mentioned categories.

Targets 1 to 3 (see Figures 3.2 to 3.4 respectively) manœuvre in the 3D space. Each corresponding figure shows (a) the target trajectory in the 3D space, (b) the target position in the horizontal plane (xy-plane), (c) the target altitude (position along the z-axis), (d) the target velocity (along each of the x, y and z axes), and (e) the target acceleration (along each of the x, y and z axes).

Targets 4 to 6 (see Figures 3.5 to 3.7 respectively) manœuvre in a horizontal plane (constant altitude). Each corresponding figure shows (a) the target position in the horizontal plane (xy-plane), (b) the target altitude (position along the z-axis), (c) the target velocity (along each of the x and y axes), and (d) the target acceleration (along each of the x and y axes).

Let L denote the total number of scans (time steps) for the duration of tracking a target (that is, the number of points on a target trajectory). The simulation scenarios are described as follows.

Target 1 ($L = 100$):

The initial state vector is $[26689, 15840, 40, -\sqrt{80}, -\sqrt{159920}, 0, 0, 0, 0]^T$. From 0 to 20 seconds, it moves at constant velocity. From 20 to 35 seconds, it makes a coordinated turn to the right. From 35 to 55 seconds, it moves at constant velocity.

From 55 to 70 seconds, it makes a coordinated turn to the left. From 70 to 99 seconds, it moves at constant velocity.

Target 2 ($L = 100$):

The initial state vector is $[3000, 5000, 1000, 100, 0, 0, 0, 0, 0]^T$. From 0 to 15 seconds, it moves at constant velocity. From 15 to 25 seconds, it makes a coordinated turn to the left. From 25 to 29 seconds, it moves at constant acceleration. From 29 to 35 seconds, it makes a coordinated turn to the right. From 35 to 40 seconds, it moves at constant velocity. From 40 to 49 seconds, it makes a coordinated turn to the right. From 49 to 54 seconds, it moves at constant acceleration. From 54 to 66 seconds, it makes a coordinated turn to the left. From 66 to 99 seconds, it moves at constant velocity.

Target 3 ($L = 100$):

The initial state vector is $[12300, 13500, 3500, -320, -85, 0, 0, 0, 0]^T$. From 0 to 12 seconds, it moves at constant velocity. From 12 to 20 seconds, it makes a coordinated turn to the left. From 20 to 27 seconds, it moves at constant velocity. From 27 to 35 seconds, it makes a coordinated turn to the right. From 35 to 59 seconds, it moves at constant velocity.

Target 4 ($L = 90$):

The initial state vector is $[21689, 10840, 40, -\sqrt{80}, -\sqrt{159920}, 0, 0, 0, 0]^T$. From 0 to 20 seconds, it moves at constant velocity. From 20 to 35 seconds, it makes a coordinated turn to the right. From 35 to 55 seconds, it moves at constant velocity. From 55 to 70 seconds, it makes a coordinated turn to the left. From 70 to 89 seconds, it moves at constant velocity.

Target 5 ($L = 80$):

The initial state vector is $[3000, 5000, 1000, 100, 0, 0, 0, 0, 0]^T$. From 0 to 5 seconds, it moves at constant velocity. From 5 to 15 seconds, it makes a coordinated turn to the left. From 15 to 19 seconds, it moves at constant acceleration. From 19 to 25 seconds, it makes a coordinated turn to the right. From 25 to 30 seconds, it moves at constant velocity. From 30 to 39 seconds, it makes a coordinated turn to the right. From 39 to 44 seconds, it moves at constant acceleration. From 44 to 56 seconds, it makes a coordinated turn to the left. From 56 to 79 seconds, it moves at constant velocity.

Target 6 ($L = 60$):

The initial state vector is $[15500, 9500, 200, -320, -85, 0, 0, 0, 0]^T$. From 0 to 7 seconds, it moves at constant velocity. From 7 to 15 seconds, it makes a coordinated turn to the left. From 15 to 22 seconds, it moves at constant velocity. From 22 to 30 seconds, it makes a coordinated turn to the right. From 30 to 59 seconds, it moves at constant velocity.

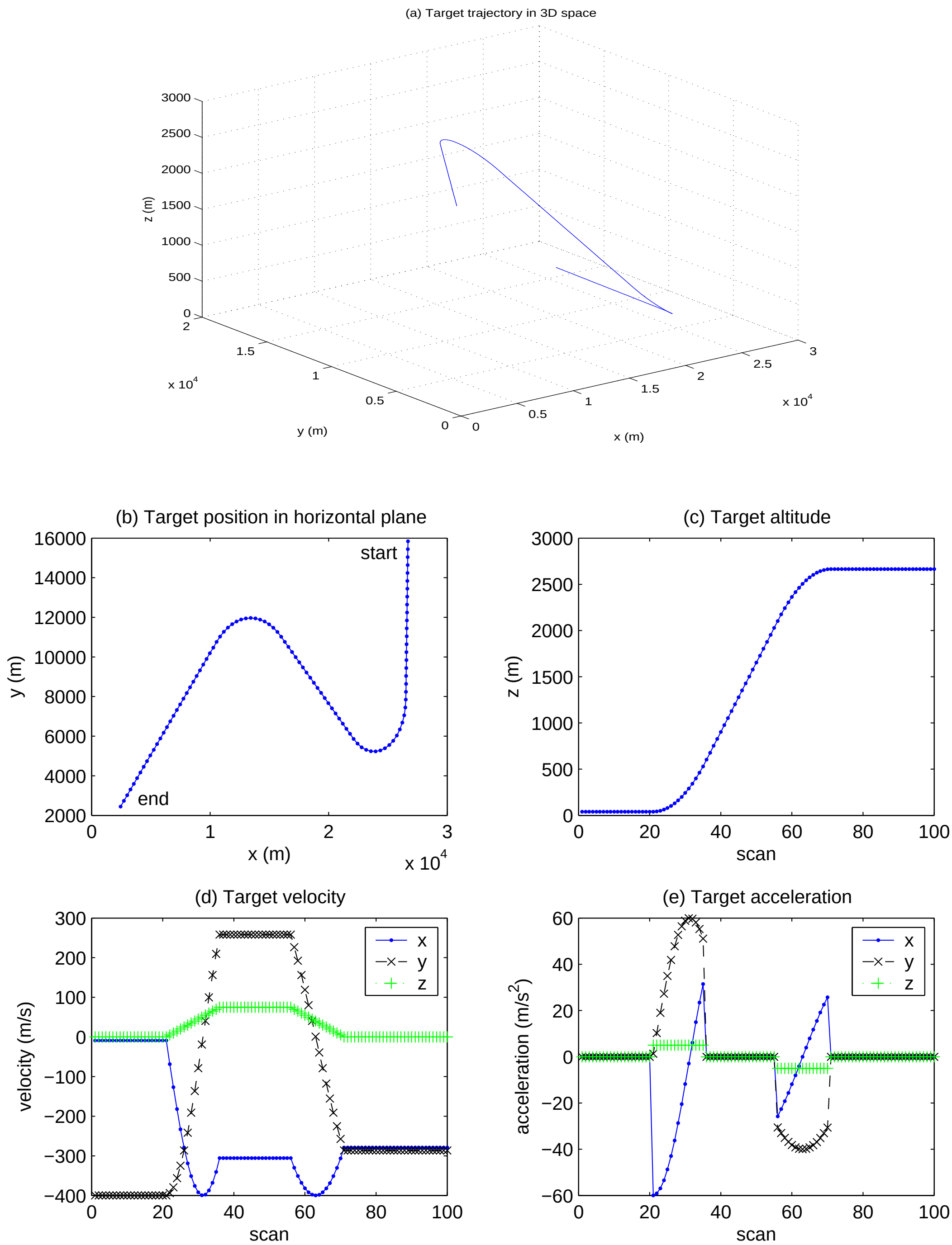

Figure 3.2: Target trajectory 1.

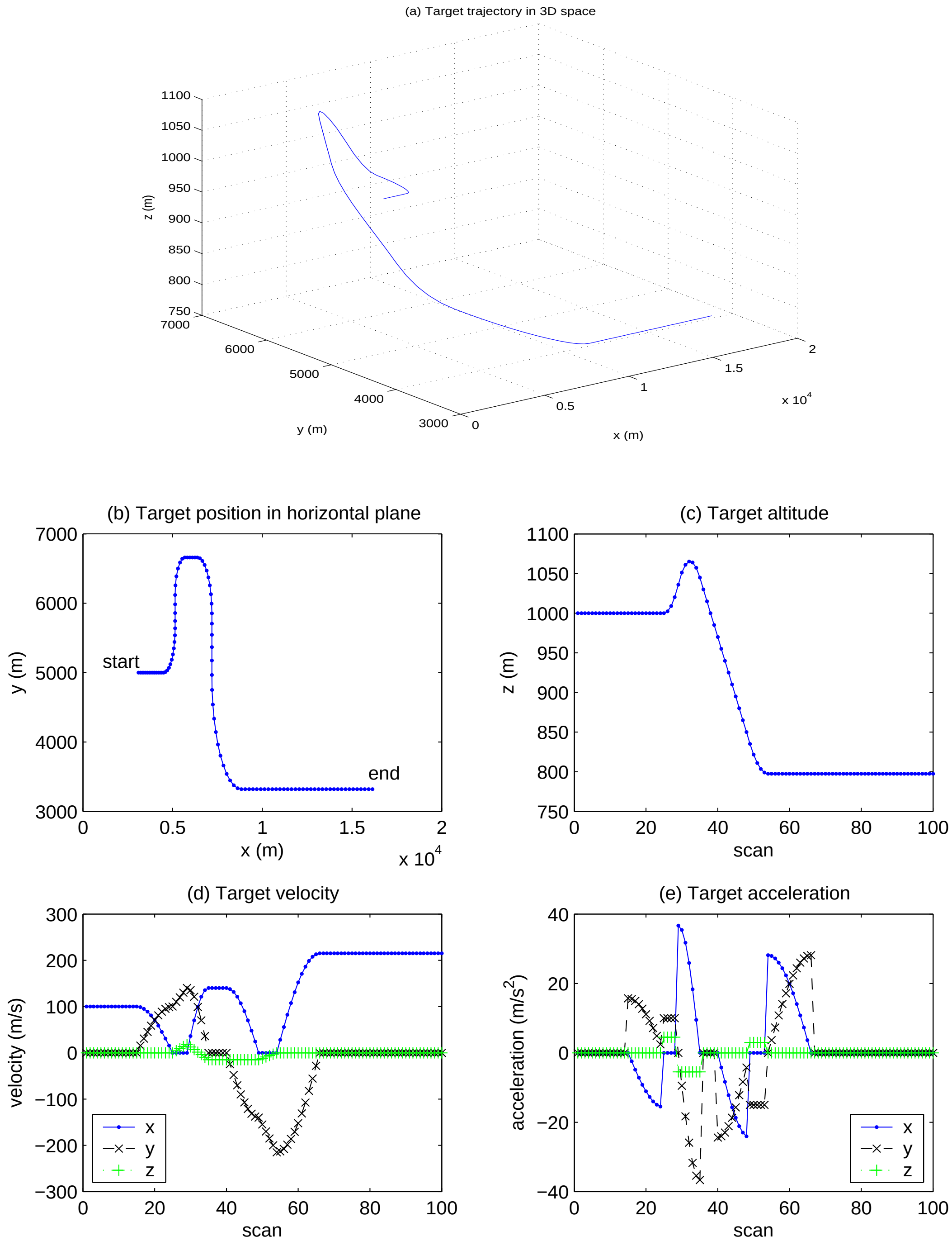

Figure 3.3: Target trajectory 2.

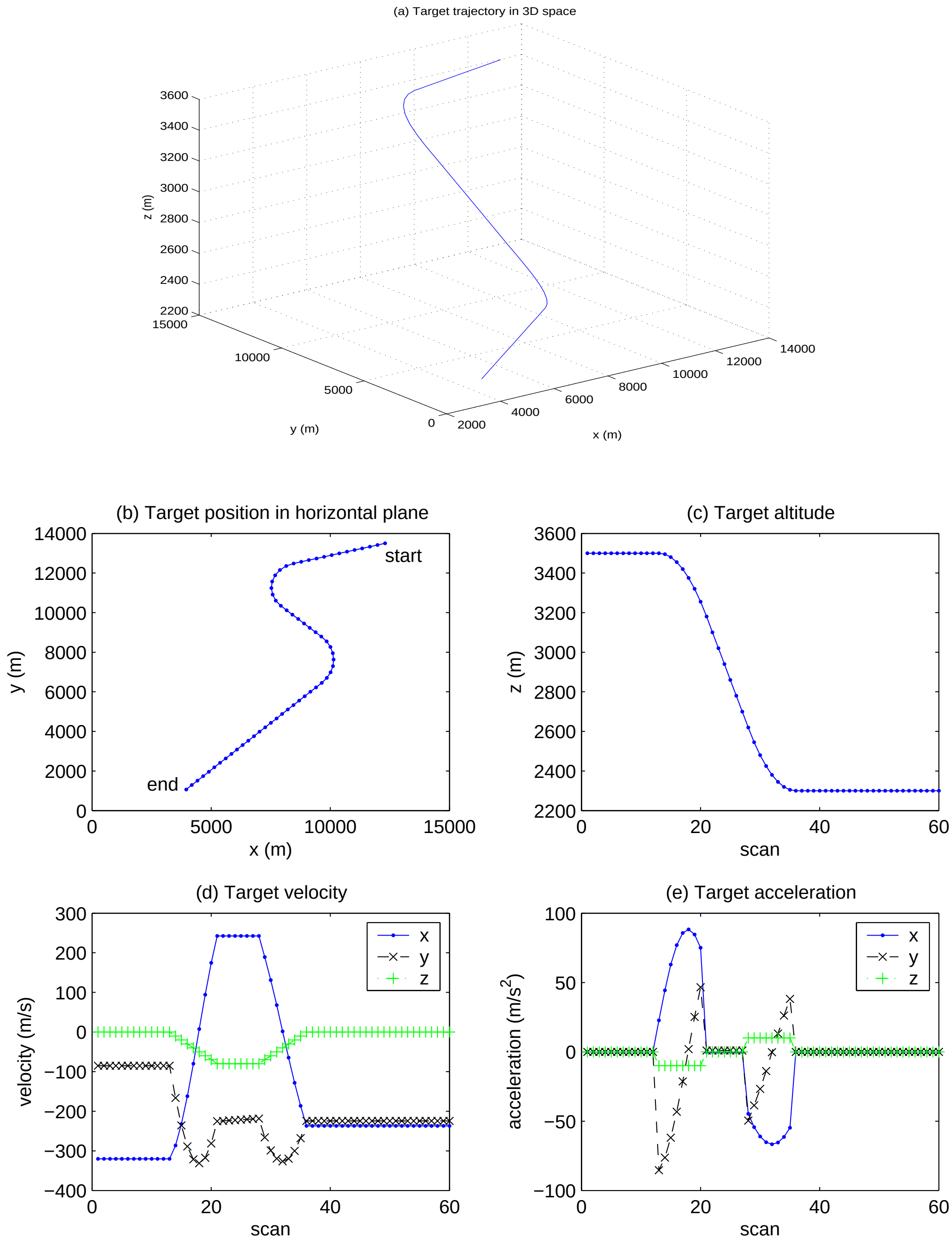

Figure 3.4: Target trajectory 3.

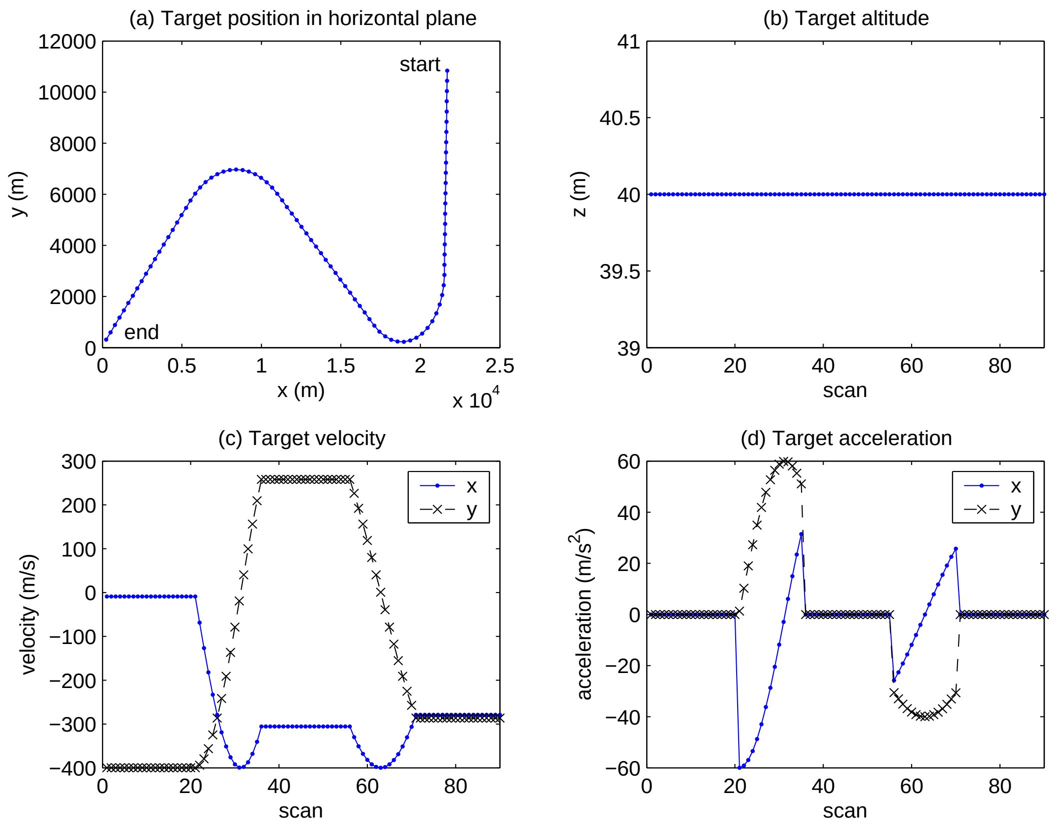

Figure 3.5: Target trajectory 4.

3.4.1.2 Computational Complexity

Let X be the notation for an arbitrary IMM algorithm variant listed in Table 3.1. The computational complexity of X is considered in the following aspects.

1. processing time (in seconds);
2. analytic time complexity;
3. relationship between processing time and analytic time complexity;
4. sensitivity of the computation time required by a filtering algorithm with respect to the sampling interval.

1. Processing time

Let t_X (respectively, $\tilde{t}_X$) denote the mean processing time per simulation run (respectively, per scan) required by algorithm X. Define τ_X (respectively, $\tilde{\tau}_X$) as the average of t_X (respectively, $\tilde{t}_X$) calculated over the six sets of simulation tests. To measure the relative computational complexity of X with respect to IEK, the ratio τ_X/τ_{IEK} (respectively, $\tilde{\tau}_X/\tilde{\tau}_{IEK}$) is computed.

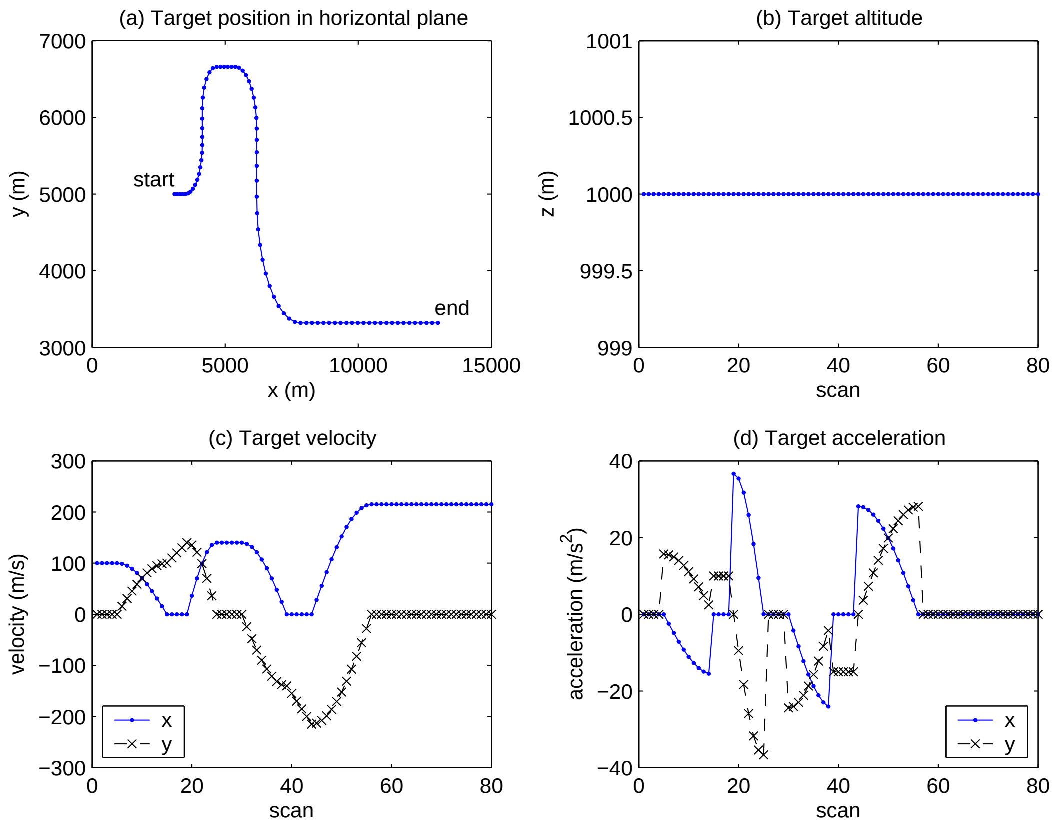

Figure 3.6: Target trajectory 5.

2. Analytic time complexity

Consider the number of operations required for one cycle of each filter and express it using the big-O notation (in terms of $r = 3$, $n_x = 9$ and $N_s = 300$). The approximate computational complexity of each filter discussed in Section 3.3 is described below.

- Extended Kalman filter and unscented Kalman filter: the two filters have comparable computational complexity, each being of order $O(n_x^3)$ [13, 82].

- Standard particle filter (Algorithm 3):

  - the importance sampling step comprises matrix-vector multiplication involving the system matrices and vectors, which requires $O(N_s(n_x^2 + n_x n_z))$ operations. As $n_z \leq n_x$, this simplifies to a computational complexity of order $O(N_s n_x^2)$;

  - the resampling procedure (systematic resampling) used in this thesis has computational complexity of order $O(N_s)$ [9];

  - the propagation/update step concerns the computation of the updated state

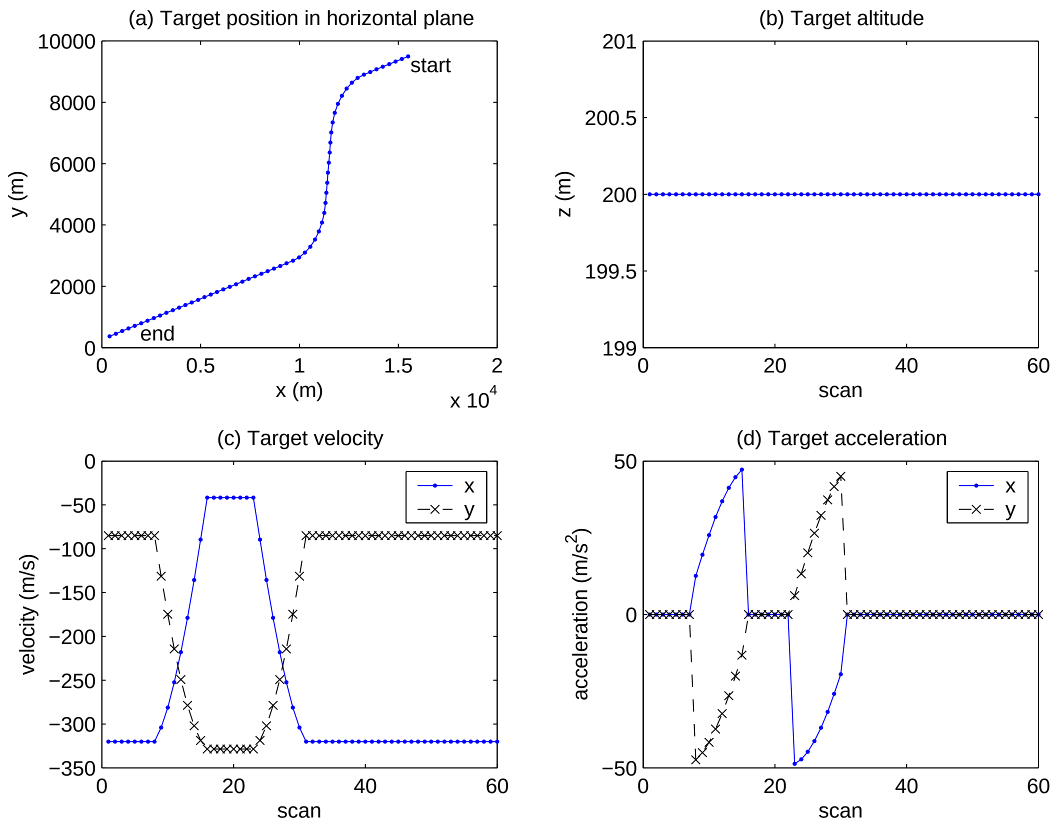

Figure 3.7: Target trajectory 6.

estimate and its corresponding state error covariance at a computational complexity of order $O(N_s(n_x + n_x^2))$, which simplifies to $O(N_s n_x^2)$.

SPF has computational complexity of order $O(N_s n_x^2)$ (by considering the dominant term).

- Auxiliary particle filter (Algorithm 4): with reference to SPF,
  - an importance sampling step which requires $O(N_s n_x^2)$ operations is added;
  - a resampling step which requires $O(N_s)$ operations is added.

  APF has computational complexity of order $O(N_s n_x^2)$ (by considering the dominant term).

- Regularized particle filter (Algorithm 5): with reference to SPF, a regularization step is added during resampling, requiring
  - generation of N_s additional samples from the kernel, and
  - $O(N_s n_x^2)$ operations for matrix-vector multiplication involving the Cholesky factor of a covariance matrix and these samples.

RPF has computational complexity of order $O(N_s n_x^2)$ (by considering the dominant term).

- Extended Kalman particle filter or unscented particle filter (Algorithm 6): with reference to SPF,
  - each of the N_s particles uses an EKF or a UKF, an $O(n_x^3)$ technique, to generate the importance density, so the importance sampling step has a higher computational complexity of order $O(N_s n_x^3)$ [320];
  - the resampling procedure requires $O(N_s)$ operations;
  - the propagation step requires $O(N_s n_x^2)$ operations.

  EKPF and UPF each has computational complexity of order $O(N_s n_x^3)$ (by considering the dominant term).

- Gaussian particle filter:
  - importance sampling comprises matrix-vector multiplication involving the system matrices and vectors, which requires $O(N_s n_x^2)$ operations;
  - computation of the predicted state estimate and its corresponding state error covariance requires $O(N_s n_x^2)$ operations;
  - computation of the updated state estimate and its corresponding state error covariance requires $O(N_s n_x^2)$ operations.

  GPF has computational complexity of order $O(N_s n_x^2)$.

The computational complexity of an IMM algorithm variant is computed as a linear combination of those of the filters required for the respective combination stated in Table 3.1:

- IEK, IUK - $O(n_x^3 + n_x^3 + n_x^3)$, that is, $O(3n_x^3)$;
- filters that use APF, GPF, RPF or SPF - $O(n_x^3 + n_x^3 + N_s n_x^2)$, which simplifies to $O(N_s n_x^2)$, as N_s is chosen to be much larger than n_x;
- filters that use EKPF or UPF - $O(n_x^3 + n_x^3 + N_s n_x^3)$, which simplifies to $O(N_s n_x^3)$.

Tables 3.2 and 3.3 show the computational complexity of each IMM algorithm variant. The last column displays the relative computational complexity of each variant with respect to IEK.

Filter X	Big-O notation	Mean processing time per simulation run (s)							
		TGT 1	TGT 2	TGT 3	TGT 4	TGT 5	TGT 6	Avg.	τ_X/τ_{IEK}
IEK	$3n_x^3$	0.223	0.212	0.122	0.193	0.167	0.116	0.17	1.00
IUK	$3n_x^3$	0.307	0.298	0.175	0.276	0.237	0.172	0.24	1.42
IEE	$N_s n_x^3$	21.062	20.977	12.714	19.125	16.740	12.389	17.17	99.77
IEG	$N_s n_x^2$	14.668	14.710	8.723	13.265	11.837	8.869	12.01	69.80
IER	$N_s n_x^2$	9.220	9.248	5.472	8.309	7.445	5.557	7.54	43.83
IES	$N_s n_x^2$	9.204	9.228	5.462	8.290	7.420	5.558	7.53	43.74
IEU	$N_s n_x^3$	42.648	42.136	25.797	39.368	33.961	25.435	34.89	202.76
IEA	$N_s n_x^2$	19.566	19.606	11.654	17.641	15.798	11.865	16.02	93.11
IUE	$N_s n_x^3$	21.055	20.975	12.728	19.133	16.769	12.403	17.18	99.82
IUG	$N_s n_x^2$	14.685	14.735	8.740	13.298	11.863	8.897	12.04	69.95
IUR	$N_s n_x^2$	9.231	9.267	5.487	8.333	7.468	5.576	7.56	43.93
IUS	$N_s n_x^2$	9.220	9.246	5.476	8.313	7.446	5.566	7.54	43.84
IUU	$N_s n_x^3$	42.598	42.177	25.797	39.357	33.954	25.426	34.88	202.72
IUA	$N_s n_x^2$	19.583	19.629	11.665	17.662	15.822	11.873	16.04	93.21
IEUE	$N_s n_x^3$	21.037	20.995	12.718	19.123	16.756	12.403	17.17	99.79
IEUG	$N_s n_x^2$	14.690	14.741	8.742	13.304	11.866	8.898	12.04	69.97
IEUR	$N_s n_x^2$	9.242	9.276	5.492	8.340	7.466	5.582	7.57	43.97
IEUS	$N_s n_x^2$	9.232	9.258	5.482	8.320	7.449	5.576	7.55	43.89
IEUU	$N_s n_x^3$	42.668	42.127	25.795	39.332	33.952	25.437	34.89	202.73
IEUA	$N_s n_x^2$	19.616	19.633	11.673	17.669	15.817	11.886	16.05	93.26

Table 3.2: Computational complexity (per simulation run).

Filter X	Big-O notation	Mean processing time per scan (s)							
		TGT 1	TGT 2	TGT 3	TGT 4	TGT 5	TGT 6	Avg.	$\tilde{\tau}_X/\tilde{\tau}_{IEK}$
IEK	$3n_x^3$	0.002	0.002	0.002	0.002	0.002	0.002	0.002	1.00
IUK	$3n_x^3$	0.003	0.003	0.003	0.003	0.003	0.003	0.003	1.42
IEE	$N_s n_x^3$	0.211	0.210	0.212	0.213	0.209	0.206	0.210	100.49
IEG	$N_s n_x^2$	0.147	0.147	0.145	0.147	0.148	0.148	0.147	70.34
IER	$N_s n_x^2$	0.092	0.092	0.091	0.092	0.093	0.093	0.092	44.15
IES	$N_s n_x^2$	0.092	0.092	0.091	0.092	0.093	0.093	0.092	44.07
IEU	$N_s n_x^3$	0.426	0.421	0.430	0.437	0.425	0.424	0.427	204.37
IEA	$N_s n_x^2$	0.196	0.196	0.194	0.196	0.197	0.198	0.196	93.85
IUE	$N_s n_x^3$	0.211	0.210	0.212	0.213	0.210	0.207	0.210	100.55
IUG	$N_s n_x^2$	0.147	0.147	0.146	0.148	0.148	0.148	0.147	70.49
IUR	$N_s n_x^2$	0.092	0.093	0.091	0.093	0.093	0.093	0.093	44.27
IUS	$N_s n_x^2$	0.092	0.092	0.091	0.092	0.093	0.093	0.092	44.17
IUU	$N_s n_x^3$	0.426	0.422	0.430	0.437	0.424	0.424	0.427	204.34
IUA	$N_s n_x^2$	0.196	0.196	0.194	0.196	0.198	0.198	0.196	93.95
IEUE	$N_s n_x^3$	0.210	0.210	0.212	0.212	0.209	0.207	0.210	100.52
IEUG	$N_s n_x^2$	0.147	0.147	0.146	0.148	0.148	0.148	0.147	70.51
IEUR	$N_s n_x^2$	0.092	0.093	0.092	0.093	0.093	0.093	0.093	44.30
IEUS	$N_s n_x^2$	0.092	0.093	0.091	0.092	0.093	0.093	0.092	44.23
IEUU	$N_s n_x^3$	0.427	0.421	0.430	0.437	0.424	0.424	0.427	204.34
IEUA	$N_s n_x^2$	0.196	0.196	0.195	0.196	0.198	0.198	0.197	94.00

Table 3.3: Computational complexity (per scan).

3. Relationship between processing time and analytic time complexity

Consider the relationship between the mean processing time per simulation run[2] and the analytic time complexity for each algorithm. Let $O(b_X)$ denote the analytic time order for algorithm X, where $b(\cdot)$ is a function of r, n_x and N_s (for instance, b_{IEK} is $3n_x^3$ (since $r = 3$), b_{IUR} is $N_s n_x^2$ and b_{IEUE} is $N_s n_x^3$). Define the ratio

$$\hat{r}_X := \frac{\tau_X}{b_X}, \tag{3.41}$$

and the corresponding normalization with respect to $\hat{r}_{IEK}$,

$$r_X := \frac{\hat{r}_X}{\hat{r}_{IEK}}. \tag{3.42}$$

It can be seen from Figure 3.8 that the ratio r_X for algorithm X is a positive constant of moderate magnitude. This implies that the processing time for each filtering algorithm is proportional to its computational complexity, in particular, the corresponding analytic time order.

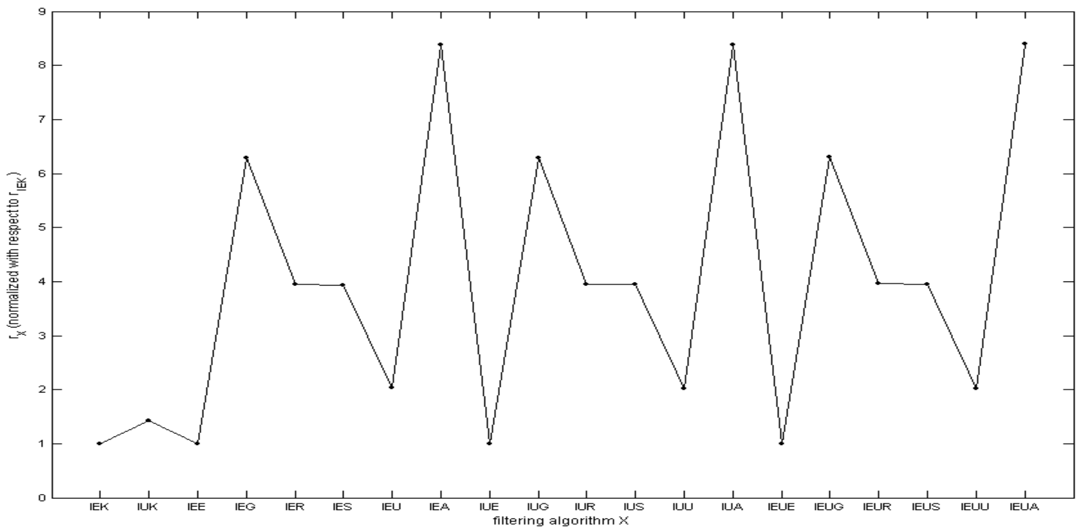

Figure 3.8: Processing time relative to analytic time complexity.

For any two arbitrary filtering algorithms X and Y, by manipulating Equations 3.41 and 3.42, we get

$$\frac{\tau_X}{\tau_Y} = \left(\frac{r_X}{r_Y}\right)\frac{b_X}{b_Y}. \tag{3.43}$$

Since r_X/r_Y is a positive constant, the relative processing time for the two algorithms is proportional to their relative computational complexity (in terms of analytic time order).

[2]Result also applies to the case of mean processing time per scan.

4. Sensitivity of the computation time with respect to the sampling interval We now study the sensitivity of the computation time required by a filtering algorithm with respect to the sampling interval $T = 1$ (second).

The mean processing time per scan required by each filtering algorithm is less than the sampling interval, ranging from less than 0.005 seconds for IEK and IUK to less than 0.5 seconds for algorithms that use unscented particle filters. Thus, for each algorithm implemented, there would be time for reaction to the target motion before the next scan.

In the event that different filtering algorithms yield results of comparable accuracy, it is apparent that those with lower computation time requirements would be preferred over those with higher requirements. For a situation in which the rate of response is crucial, much emphasis would be placed on the computation time required by the filtering algorithm used. It would be ideal to be able to minimize the amount of processing time, without the accuracy of results being compromised. An example is the case of tracking an aircraft in military or civilian air traffic surveillance. It is of paramount importance for the decision makers (military defenders or air traffic controllers) to derive and predict the pilot's intent accurately as fast as possible, as well as to have sufficient time for reaction to the aircraft motion.

It is also noted from additional numerical tests that, when the number of particles used in a particle filter is increased from the current $N_s = 300$ to $N_s = 600$, the corresponding filtering algorithm requires about twice the current amount of computation time as well. In particular, it becomes infeasible to use an IMM algorithm which uses an unscented particle filter because the computation time required would then be close to the sampling interval. This would leave the decision makers with almost no time for reaction.

3.4.1.3 Analysis of Numerical Results

Analysis of the simulation test results is carried out in three parts:

1. comparison of state estimation errors obtained with the different filtering algorithms;
2. statistical comparison of the filtering algorithms, formulated as a hypothesis problem;
3. consistency of the filtering algorithms.

1. Comparison of state estimation errors

The *root mean square error* (RMSE) in the estimation of position (respectively, velocity and acceleration) is defined as

$$\text{RMSE} := \left\{ \frac{1}{C} \frac{1}{L} \sum_{i=1}^{C} \sum_{k=1}^{L} \left\| \hat{u}_{ik} - u_k \right\|_2^2 \right\}^{\frac{1}{2}}, \tag{3.44}$$

where

- $\hat{u}_{ik}$ is the position (respectively, velocity and acceleration) estimate at the k-th scan in the i-th simulation run, $i = 1, \ldots, C$, $k = 1, \ldots, L$, and
- u_k is the actual target position (respectively, velocity and acceleration) at the k-th scan, $k = 1, \ldots, L$.

Table 3.4 shows the RMSEs (in metres) in position estimation with measurement data for the target trajectory in each of the simulation scenarios. Tables 3.5 to 3.7 show the RMSEs in state (namely, position, velocity and acceleration) estimation for the IMM algorithm variants implemented. For each algorithm used, consider the average of RMSEs in state estimation computed over the six sets of simulation tests. For each state, ε_X denotes the average RMSE in state estimation for algorithm X. The ratio $\varepsilon_X/\varepsilon_{IEK}$ is computed as a measure of the relative average RMSE in state estimation for algorithm X with respect to IEK. An entry "-" indicates unavailability due to occurrence(s) of divergence of the corresponding algorithm. The notation "$O(10^n)$" (with n being a relevant positive integer) is used to represent very large errors in state estimation.

Target trajectory	1	2	3	4	5	6
RMSE (in metres, to 2 decimal places)	773.79	778.45	779.70	772.75	773.26	774.81

Table 3.4: RMSE in position estimation with measurement data

We identify the filtering algorithms that yield smaller estimation errors than those obtained with IEK for each state as follows.

Filter X	RMSE in position estimation (to 2 decimal places)							
	TGT 1	TGT 2	TGT 3	TGT 4	TGT 5	TGT 6	Avg.	$\varepsilon_X/\varepsilon_{IEK}$
IEK	473.69	455.42	483.50	473.33	459.77	466.86	468.76	1.00
IUK	470.87	438.38	483.72	471.79	447.07	466.14	463.00	0.99
IEE	664.26	-	629.81	664.09	-	627.85	-	-
IEG	-	525.89	-	567.25	521.17	549.54	-	-
IER	478.22	453.90	487.16	476.34	460.23	469.87	470.95	1.00
IES	474.68	453.39	485.20	475.18	456.85	467.14	468.74	1.00
IEU	-	798.02	763.93	811.85	778.65	753.00	-	-
IEA	515.76	479.92	508.85	514.21	475.09	498.90	498.79	1.06
IUE	665.46	612.47	629.18	667.46	-	627.71	-	-
IUG	565.14	528.00	-	562.55	519.56	543.03	-	-
IUR	467.27	439.23	476.00	468.04	452.25	460.12	460.49	0.98
IUS	465.44	437.21	475.25	465.95	451.52	458.74	459.02	0.98
IUU	819.99	-	754.80	812.39	772.77	750.22	-	-
IUA	513.55	470.70	502.83	511.69	470.50	495.52	494.13	1.05
IEUE	663.52	-	627.51	663.20	-	625.81	-	-
IEUG	-	526.49	551.99	-	520.00	-	-	-
IEUR	475.99	452.37	485.78	476.03	460.07	468.25	469.75	1.00
IEUS	477.71	451.47	483.50	474.34	461.57	468.95	469.59	1.00
IEUU	-	-	758.79	809.38	774.72	756.43	-	-
IEUA	515.65	476.30	508.88	515.86	479.14	501.38	499.53	1.07

Table 3.5: Errors in position estimation.

- position - IUK, IUR, IUS: almost identical;
- velocity - IUK, IUR, IUS: about 10% less,
  - IEUR, IEUS: about 3% less;
- acceleration - IUR, IUS, IEUR, IEUS: about 50% less;
  - IER, IES, IEA, IUA, IEUA: about 25–35% less.

IEK yields comparable results to IUR, IUS, IEUR and IEUS in position estimation, but is less effective in the estimation of velocity and acceleration. It is noted that IEK requires much lower (about 45 times less) computational cost than the four algorithms which use particle filters. It is also noted that IUK yields comparatively smaller errors in position and velocity estimation, but diverges in acceleration estimation. This detracts it from being a suitable algorithm for the current problem.

For the remaining IMM algorithms, excessively large errors in state estimation or divergence are obtained in the simulation tests. It can be seen from the tabulated results that position (respectively, velocity) estimation errors obtained with IMM algorithms which use APF are less than 10% (respectively, 20%) larger than that obtained with

Filter X	RMSE in velocity estimation (to 2 decimal places)							
	TGT 1	TGT 2	TGT 3	TGT 4	TGT 5	TGT 6	Avg.	$\varepsilon_X/\varepsilon_{IEK}$
IEK	165.59	154.79	193.06	175.38	165.91	175.66	171.73	1.00
IUK	157.90	127.74	185.62	162.21	136.94	159.76	155.03	0.90
IEE	729.96	-	644.66	716.89	-	614.48	-	-
IEG	-	225.67	-	290.26	218.95	271.24	-	-
IER	174.66	150.33	198.85	176.34	161.88	177.31	173.23	1.01
IES	172.98	152.67	200.45	174.81	155.35	169.60	170.98	1.00
IEU	-	$O(10^8)$	$O(10^5)$	$O(10^8)$	$O(10^7)$	$O(10^5)$	-	-
IEA	208.99	177.04	214.60	210.82	176.52	204.02	198.66	1.16
IUE	731.45	603.03	620.36	714.89	-	607.09	-	-
IUG	289.90	231.03	-	288.99	219.99	260.99	-	-
IUR	158.75	132.94	184.65	161.37	143.69	157.27	156.45	0.91
IUS	157.34	131.68	184.57	160.07	143.46	156.49	155.60	0.91
IUU	$O(10^9)$	-	$O(10^5)$	$O(10^8)$	$O(10^7)$	$O(10^5)$	-	-
IUA	202.41	167.77	205.70	205.47	167.29	197.53	191.03	1.11
IEUE	725.10	-	617.13	748.80	-	610.38	-	-
IEUG	-	223.74	270.16	-	228.32	-	-	-
IEUR	165.99	145.75	193.17	170.73	154.95	172.11	167.12	0.97
IEUS	167.73	145.74	191.89	171.89	154.96	167.02	166.54	0.97
IEUU	-	-	$O(10^5)$	$O(10^8)$	$O(10^7)$	$O(10^5)$	-	-
IEUA	207.73	170.94	210.96	210.38	178.91	202.43	196.89	1.15

Table 3.6: Errors in velocity estimation.

IEK. For IMM algorithms which use EKPF, UPF or GPF, there are occurrence(s) of divergence during state estimation. Several possible reasons for these observations are briefly discussed below.

Firstly, we recall from Section 3.3.3.6 (respectively, Section 3.3.3.7) that each particle in the framework of EKPF (respectively, UPF) uses an EKF (respectively, a UKF) to obtain a Gaussian approximation to the optimal proposal distribution. The aim is to gain more overlap between the proposal distribution and the posterior pdf. This requires a Gaussian assumption on the form of the posterior pdf. On the other hand, it has been discussed in Section 3.3.3.8 that GPF approximates the posterior pdf by a single Gaussian distribution. These filters are unlikely to work well due to the prominence of nonlinearity and/or non-Gaussianity in the current problem. Secondly, large state estimation errors or divergence for filters could be due to the accumulation of rounding errors brought about by the large number of mathematical operations required for high-dimensional problems, such as the current one (here, $n_x = 9$) [82]. Thirdly, it is also possible that the number of particles N_s used is not large enough to attain stability. Taking these factors into consideration, it can be deduced that implementing particle

Filter X	RMSE in acceleration estimation (to 2 decimal places)							
	TGT 1	TGT 2	TGT 3	TGT 4	TGT 5	TGT 6	Avg.	$\varepsilon_X/\varepsilon_{IEK}$
IEK	47.80	160.13	58.67	61.61	66.04	53.31	74.59	1.00
IUK	$O(10^{12})$	$O(10^{10})$	$O(10^{11})$	$O(10^{12})$	$O(10^{12})$	$O(10^{13})$	$O(10^{13})$	$O(10^{11})$
IEE	$O(10^{3})$	-	$O(10^{3})$	$O(10^{3})$	-	$O(10^{3})$	-	-
IEG	-	101.68	-	81.53	90.90	94.61	-	-
IER	44.80	33.27	61.64	46.53	49.70	62.31	49.71	0.67
IES	55.59	45.91	60.58	48.17	37.55	45.11	48.82	0.65
IEU	-	$O(10^{9})$	$O(10^{6})$	$O(10^{9})$	$O(10^{8})$	$O(10^{6})$	-	-
IEA	55.54	48.40	64.42	65.69	47.90	57.77	56.62	0.76
IUE	$O(10^{3})$	$O(10^{3})$	$O(10^{3})$	$O(10^{3})$	-	$O(10^{3})$	-	-
IUG	85.56	107.88	-	81.94	85.45	79.80	-	-
IUR	36.09	27.21	52.62	37.15	29.01	36.38	36.41	0.49
IUS	36.11	26.61	53.14	37.32	29.33	36.53	36.51	0.49
IUU	$O(10^{10})$	-	$O(10^{6})$	$O(10^{9})$	$O(10^{8})$	$O(10^{6})$	-	-
IUA	52.92	42.03	58.28	59.29	39.63	51.63	50.63	0.68
IEUE	$O(10^{3})$	-	$O(10^{3})$	$O(10^{3})$	-	$O(10^{3})$	-	-
IEUG	-	90.86	87.23	-	94.47	-	-	-
IEUR	36.55	28.36	53.37	37.81	29.99	37.59	37.28	0.50
IEUS	37.01	28.17	53.49	37.99	30.49	37.32	37.41	0.50
IEUU	-	-	$O(10^{6})$	$O(10^{9})$	$O(10^{8})$	$O(10^{6})$	-	-
IEUA	54.12	40.21	60.52	55.03	43.53	57.15	51.76	0.69

Table 3.7: Errors in acceleration estimation.

filters (especially the computationally intensive ones) for the IMM algorithm variants would probably incur prohibitive costs when solving larger problems, such as those with n_x or N_s increased.

2. Statistical comparison of filtering algorithms

Besides comparing the errors in state estimation, it is necessary to conduct a statistical analysis of the simulation results. Comparison of the performance (in terms of the mean square error in state estimation) of the algorithms can be formulated as a hypothesis testing problem as follows [13, Sections 1.5 and 11.5].

Let X be an arbitrary filtering algorithm implemented in the preceding simulation tests. Let J_X denote the actual mean square error in state estimation. The hypothesis testing problem is to test the null hypothesis

$$H_0 : \Delta = J_{IEK} - J_X \leq 0 \quad \text{(algorithm } X \text{ not better than IEK)}, \tag{3.45}$$

versus the alternate hypothesis

$$H_1 : \Delta = J_{IEK} - J_X > 0 \quad \text{(algorithm } X \text{ better than IEK)}, \tag{3.46}$$

subject to

$$\text{Prob}\{\text{accept } H_1 | H_0 \text{ is true}\} = \alpha \quad \text{(level of significance for hypothesis } H_0\text{)}. \tag{3.47}$$

Let $(\hat{u}_X)_{ik}$ denote the state estimate for algorithm X at the k-th scan in the i-th simulation run, $i = 1, \ldots, C$, $k = 1, \ldots, L$, and u_k be the actual state at the k-th scan, $k = 1, \ldots, L$. The statistical test is based on the mean square error differences,

$$(\Delta_X)_i := \frac{1}{L} \sum_{k=1}^{L} \left\{ \left\| (\hat{u}_{IEK})_{ik} - u_k \right\|_2^2 - \left\| (\hat{u}_X)_{ik} - u_k \right\|_2^2 \right\}, \quad i = 1, \ldots, C, \tag{3.48}$$

which are independent from run to run (that is, $(\Delta_X)_i$ is independent of $(\Delta_X)_j$ for all $j \neq i$). The test uses the sample mean of the above differences,

$$\bar{\Delta}_X := \frac{1}{C} \sum_{i=1}^{C} (\Delta_X)_i, \tag{3.49}$$

and its associated standard deviation,

$$\sigma_{\bar{\Delta}_X} := \left\{ \frac{1}{C^2} \sum_{i=1}^{C} \left((\Delta_X)_i - \bar{\Delta}_X \right)^2 \right\}^{\frac{1}{2}}. \tag{3.50}$$

The test is for zero or negative mean (H_0) versus positive mean (H_1). If the estimated mean $\bar{\Delta}_X$ is positive and statistically significant, then the alternate hypothesis H_1 is accepted.

By the application of the Central Limit Theorem (See Theorem A.1), $\bar{\Delta}_X$ can be approximated by a Gaussian (normal) distribution. Let ρ_0 represent the point on the standard Gaussian distribution corresponding to the upper tail probability of α (from Equation 3.47). If the test statistic,

$$\rho := \frac{\bar{\Delta}_X}{\sigma_{\bar{\Delta}_X}} > \rho_0, \tag{3.51}$$

then $\bar{\Delta}_X$ is accepted as positive and statistically significant. In this thesis, we shall use $\alpha = 0.05$ (that is, 5% level of significance), with threshold $\rho_0 = 1.65$. Therefore, H_1 is accepted if $\rho > 1.65$.

Tables 3.8 to 3.10 show the test for differences of mean square errors in state estimation between IEK and each of the other IMM algorithm variants. We first note that in the case of acceleration estimation for Target 2, none of the filtering algorithms implemented has improvement that is sufficient to reject the null hypothesis H_0 (at 5% level of significance).

For the other results, it can be seen from the tables that IUR and IUS have statistically significant improvement over IEK (that is, test statistic $\rho > 1.65$) in position

Filter X	Test statistic for position estimation					
	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6
IUK	1.31	11.08	-0.07	0.65	7.33	0.40
IEE	-55.78	-	-34.14	-50.69	-	-34.14
IEG	-	-26.04	-	-27.32	-20.90	-23.58
IER	-2.48	0.81	-1.53	-1.79	-0.21	-1.25
IES	-0.60	0.96	-0.76	-1.16	1.45	-0.15
IEU	-	-73.77	-55.18	-72.70	-63.68	-54.79
IEA	-18.91	-12.43	-9.47	-18.35	-6.60	-11.25
IUE	-60.67	-44.95	-36.83	-52.96	-	-35.01
IUG	-32.72	-25.97	-	-30.92	-20.36	-19.79
IUR	4.93	10.13	4.87	4.46	4.19	3.73
IUS	6.26	11.31	5.30	6.38	4.62	4.52
IUU	-77.56	-	-58.95	-69.19	-63.95	-54.76
IUA	-19.34	-7.16	-7.40	-16.30	-4.93	-9.66
IEUE	-53.66	-	-32.29	-52.26	-	-32.97
IEUG	-	-29.71	-20.81	-	-22.38	-
IEUR	-1.43	1.64	-1.28	-2.08	-0.15	-0.62
IEUS	-2.24	1.87	0.00	-0.69	-0.76	-0.93
IEUU	-	-	-48.67	-78.51	-65.86	-53.47
IEUA	-19.00	-10.00	-8.63	-18.91	-8.04	-11.82

Table 3.8: Comparison of IEK with other IMM variants in position estimation.

and velocity state estimation. IEUR and IEUS have statistically significant improvement over IEK in velocity and acceleration estimation. IUK has statistically significant improvement over IEK in velocity estimation. Overall, IEUR, IEUS and IUK do not perform as well as IUR and IUS, as they do not have significant improvement over IEK in as many cases of state estimation. The remaining algorithms have even fewer or no cases of statistically significant improvement over IEK.

Based on the statistical comparison of the filtering algorithms, it can be inferred that, in terms of state estimation errors, IUR and IUS have better performance than IEK.

3. Consistency of filtering algorithms

The consistency of a filtering algorithm X is determined by comparing the state estimation errors obtained with X and the corresponding filter-calculated state error covariances.

For each target trajectory from Section 3.4.1.1, define the RMSE in state estimation for algorithm X at the k-th scan as

$$\mathrm{RMSE}_X(k) := \left\{ \frac{1}{C} \sum_{i=1}^{C} \left\| (\hat{u}_X)_{ik} - u_k \right\|_2^2 \right\}^{\frac{1}{2}}, \tag{3.52}$$

Filter X	Test statistic for velocity estimation					
	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6
IUK	4.17	9.69	2.84	4.40	6.46	3.60
IEE	-25.42	-	-9.83	-33.61	-	-30.99
IEG	-	-17.91	-	-32.29	-12.59	-19.01
IER	-4.07	1.40	-2.11	-0.29	0.76	-0.23
IES	-2.69	0.59	-1.67	0.17	2.41	1.34
IEU	-	-2.29	-2.78	-3.00	-1.68	-4.42
IEA	-17.05	-7.31	-7.02	-8.92	-2.41	-6.24
IUE	-25.30	-39.57	-28.74	-36.80	-	-23.66
IUG	-41.57	-12.24	-	-29.70	-11.93	-16.28
IUR	4.27	7.53	5.11	4.96	5.12	4.10
IUS	5.19	8.08	5.11	5.44	5.16	4.27
IUU	-1.77	-	-3.80	-2.79	-1.93	-2.04
IUA	-16.49	-4.28	-5.28	-8.85	-0.31	-4.85
IEUE	-36.03	-	-25.04	-10.45	-	-29.09
IEUG	-	-17.23	-19.87	-	-10.76	-
IEUR	-0.21	3.04	-0.05	1.38	2.48	0.52
IEUS	-1.13	3.00	0.52	0.91	2.37	1.67
IEUU	-	-	-2.81	-1.75	-1.21	-2.77
IEUA	-16.22	-4.54	-6.57	-11.64	-2.86	-5.40

Table 3.9: Comparison of IEK with other IMM variants in velocity estimation.

Filter X	Test statistic for acceleration estimation					
	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6
IUK	-1.02	-1.01	-1.01	-1.30	-1.01	-1.01
IEE	-4.35	-	-1.50	-5.32	-	-4.84
IEG	-	0.67	-	-3.29	-2.91	-4.53
IER	0.82	1.08	-1.06	2.05	1.45	-0.55
IES	-1.00	1.04	-0.93	1.79	2.63	2.11
IEU	-	-2.23	-4.45	-3.32	-1.19	-3.40
IEA	-2.54	1.03	-2.16	-0.40	1.82	-1.17
IUE	-3.77	-5.41	-1.54	-3.56	-	-4.77
IUG	-7.27	0.61	-	-3.36	-2.30	-5.62
IUR	3.15	1.10	3.96	3.15	3.18	4.21
IUS	3.14	1.10	3.63	3.13	3.17	4.19
IUU	-2.43	-	-4.30	-2.37	-1.93	-2.92
IUA	-1.53	1.05	0.25	0.32	2.50	0.48
IEUE	-2.61	-	-3.67	-1.51	-	-3.05
IEUG	-	0.76	-7.47	-	-2.93	-
IEUR	3.05	1.09	3.47	3.08	3.13	3.97
IEUS	2.92	1.10	3.42	3.06	3.10	4.03
IEUU	-	-	-3.17	-2.85	-1.27	-3.09
IEUA	-1.93	1.06	-1.06	0.98	2.25	-0.65

Table 3.10: Comparison of IEK with other IMM variants in acceleration estimation.

where $(\hat{u}_X)_{ik}$ is the state estimate for algorithm X in the i-th simulation run, $i = 1, \ldots, C$, and u_k is the actual state, $k = 1, \ldots, L$.

Let $(\hat{P}_X)_{ik}$ denote the state error covariance corresponding to $(\hat{u}_X)_{ik}$, $i = 1, \ldots, C$, $k = 1, \ldots, L$. Define the variable

$$\mathrm{RMP}_X(k) := \left\{ \frac{1}{C} \sum_{i=1}^{C} \mathrm{trace}((\hat{P}_X)_{ik}) \right\}^{\frac{1}{2}}, \tag{3.53}$$

where trace(M) is the sum of the diagonal elements of matrix M.

A comparison of state estimation errors (Equation 3.52) with filter-calculated state error covariances (Equation 3.53) is done for the filtering algorithms that have been classified as having better performance than IEK in state estimation. In this test problem, these algorithms are IUR and IUS. Figures 3.9 to 3.14 provide graphical representation of the state estimation errors (label used: RMSE) and filter-calculated state error covariances (label used: RMP). The observations for each state are described below.

- position (Figures 3.9 and 3.10):
  RMSE generally follows the trend of RMP. RMSE is larger than RMP for the test trajectories (phenomenon known as "optimistic" [12, 13]). At the onset of a manœuvre, a surge occurs for each of the two parameters. The difference between them is larger during periods of manœuvre than during non-manœuvring periods.

- velocity (Figures 3.11 and 3.12):
  In most cases, RMSE is almost commensurate with RMP during non-manœuvring motion. In addition, RMSE is usually larger than RMP during periods of manœuvres. A surge in RMSE occurs at the start of manœuvre such that it increases till it exceeds RMP. The difference between the two parameters is also larger during periods of manœuvres.

- acceleration (Figures 3.13 and 3.14):
  RMSE generally follows the trend of RMP. RMSE is smaller than RMP (phenomenon known as "pessimistic" [12,13]) most of the time throughout tracking. At the start of a manœuvre, RMSE increases and exceeds RMP during the manœuvring period.

Based on the first two parts on analysis of the numerical results, IUR and IUS perform better than IEK in state estimation. However, since not all the state estimation errors are commensurate with the filter-calculated state error covariances, they are not entirely consistent. It has been reported that this is not an unusual phenomenon for adaptive algorithms [12], with timely adaptation being driven by the inconsistency. In practice, it is of interest to strive to make the filter as close to being consistent as feasible [13].

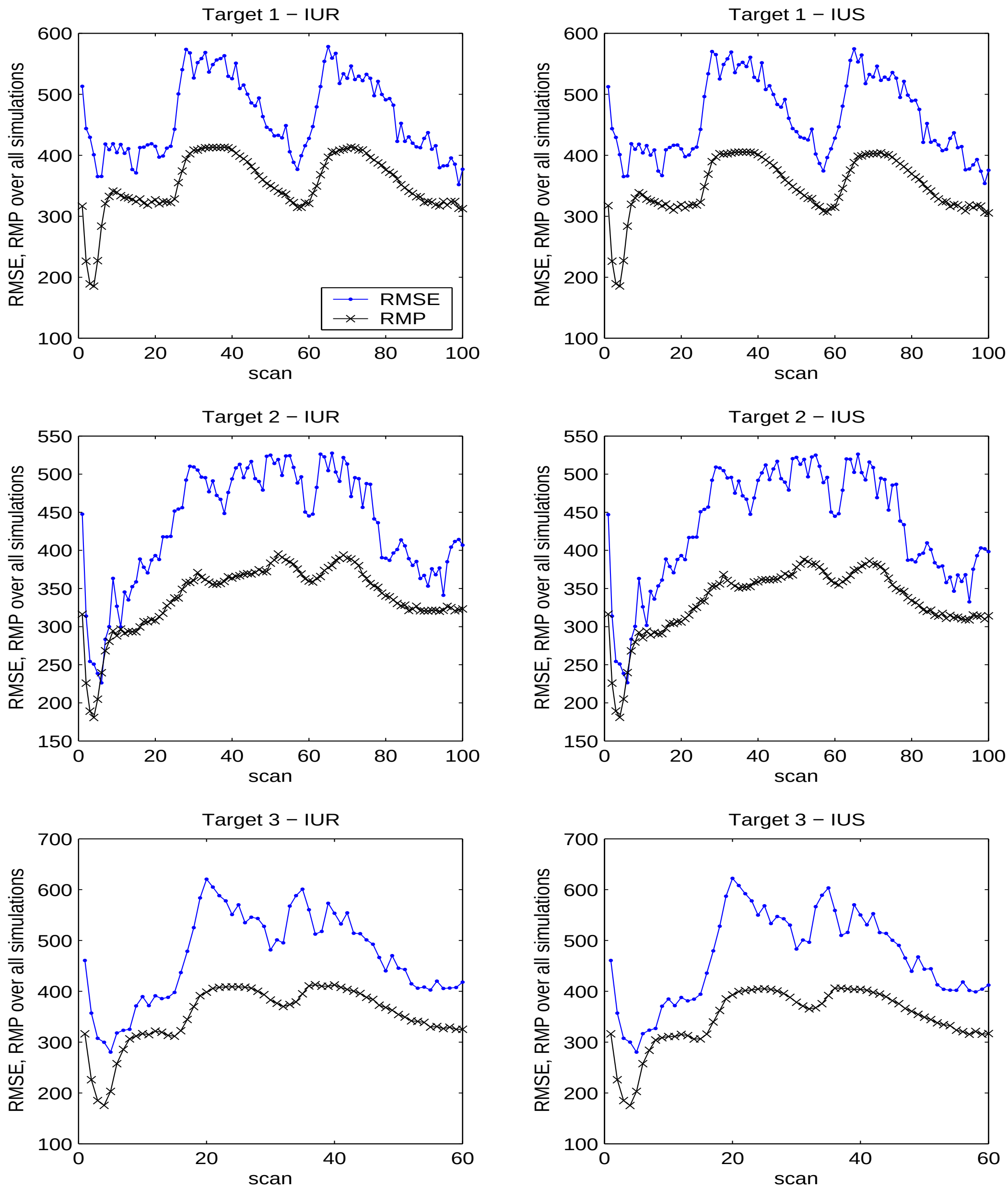

Figure 3.9: Targets 1 to 3 - Comparison of RMSE and RMP in position estimation.

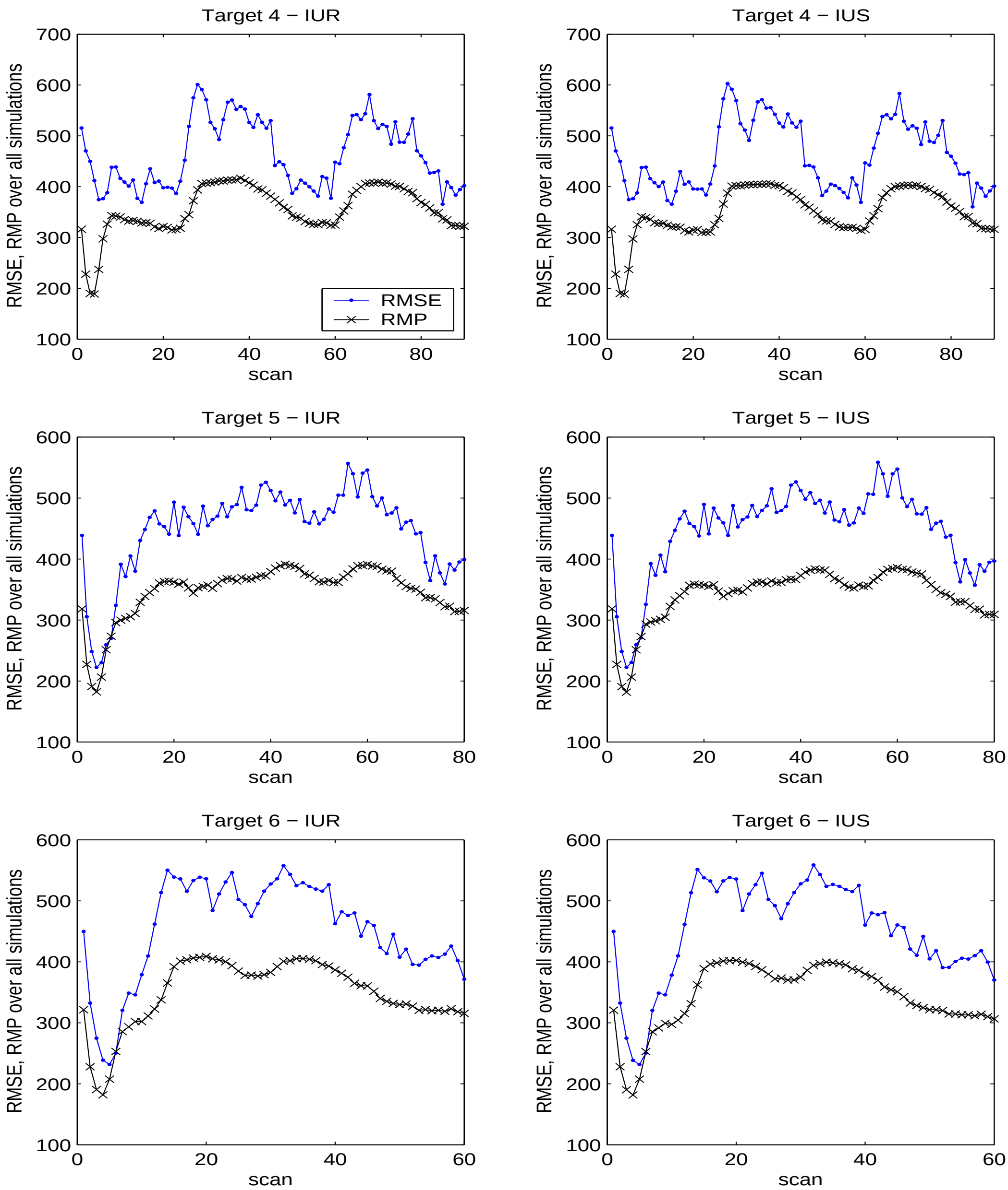

Figure 3.10: Targets 4 to 6 - Comparison of RMSE and RMP in position estimation.

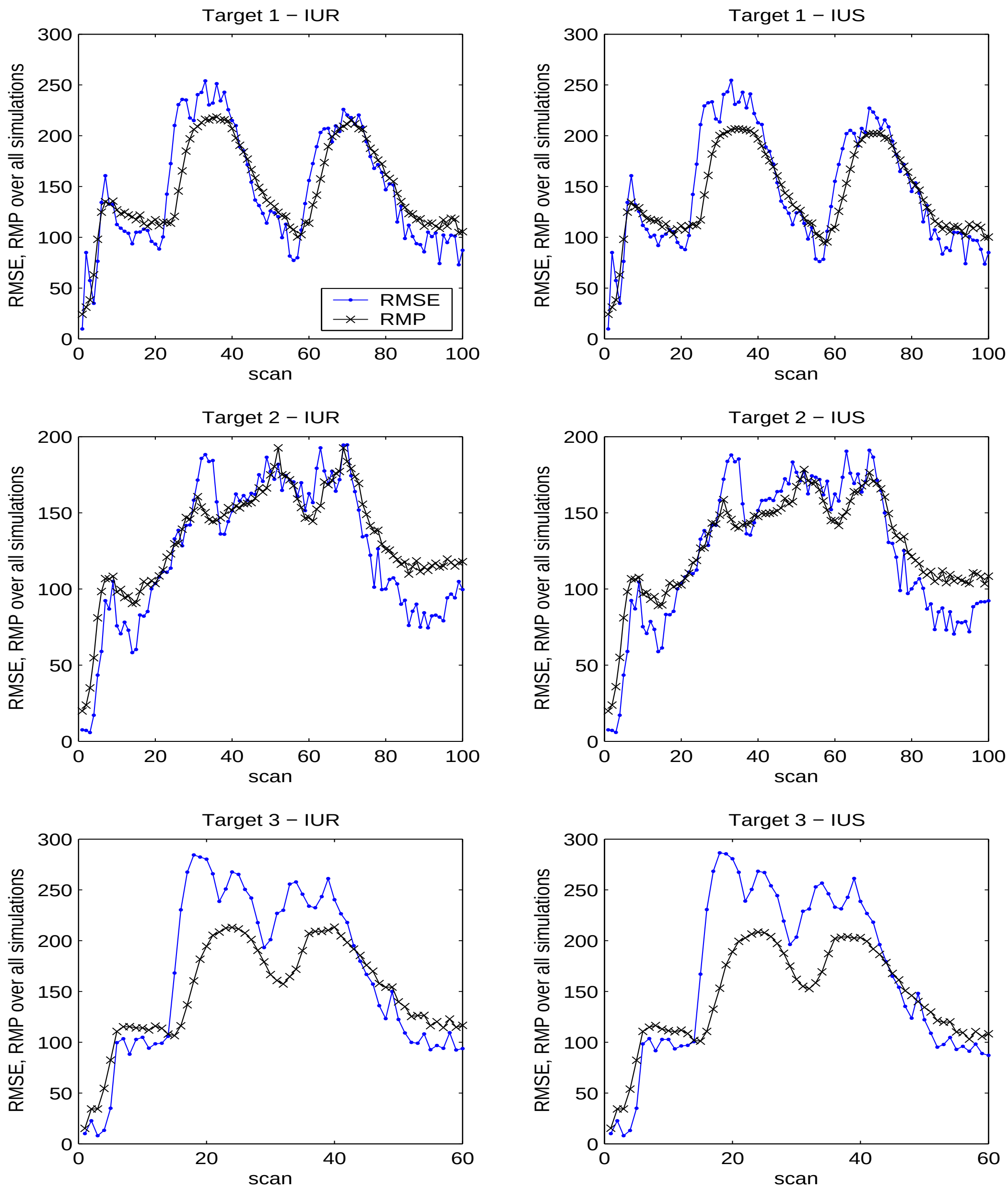

Figure 3.11: Targets 1 to 3 - Comparison of RMSE and RMP in velocity estimation.

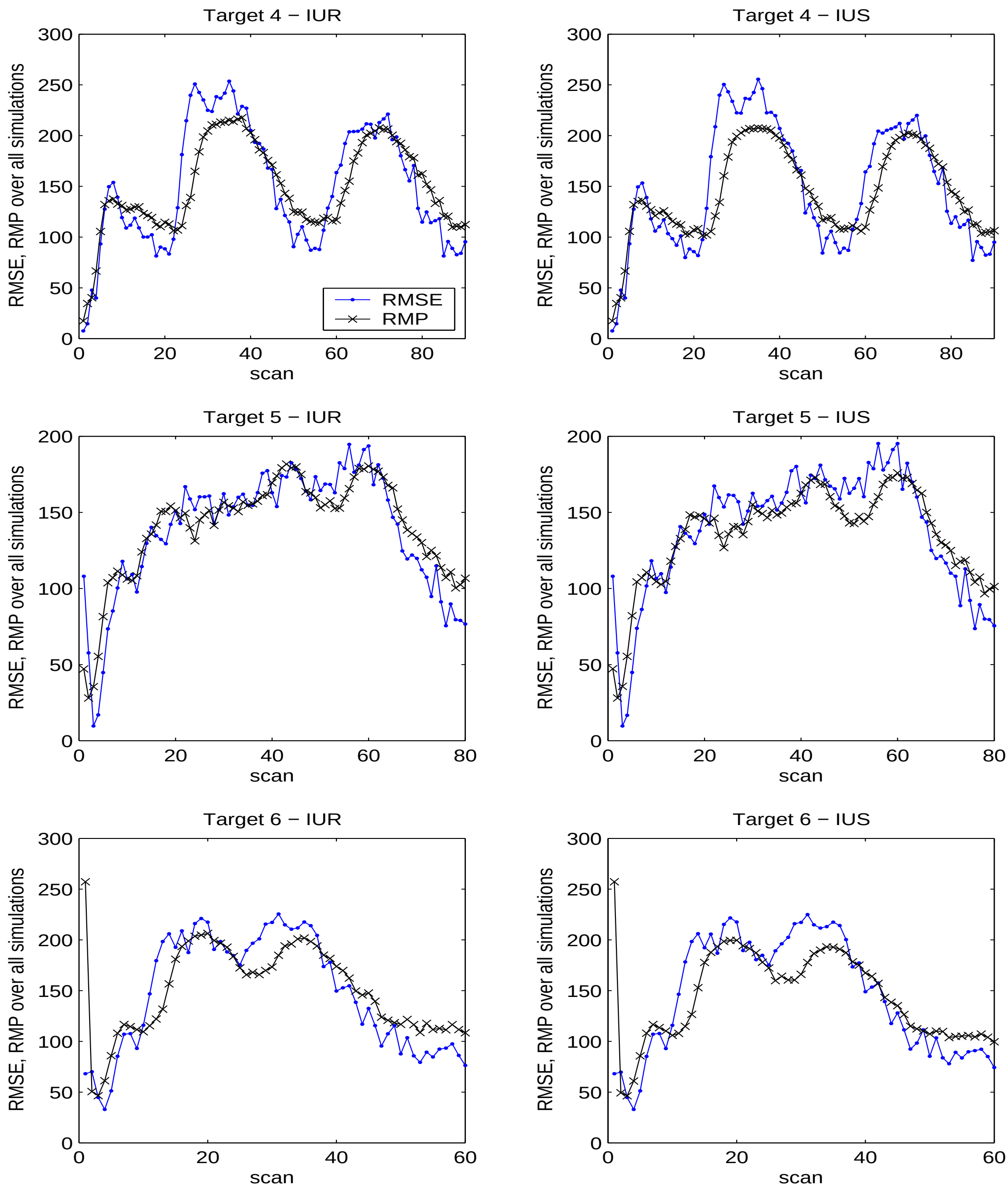

Figure 3.12: Targets 4 to 6 - Comparison of RMSE and RMP in velocity estimation.

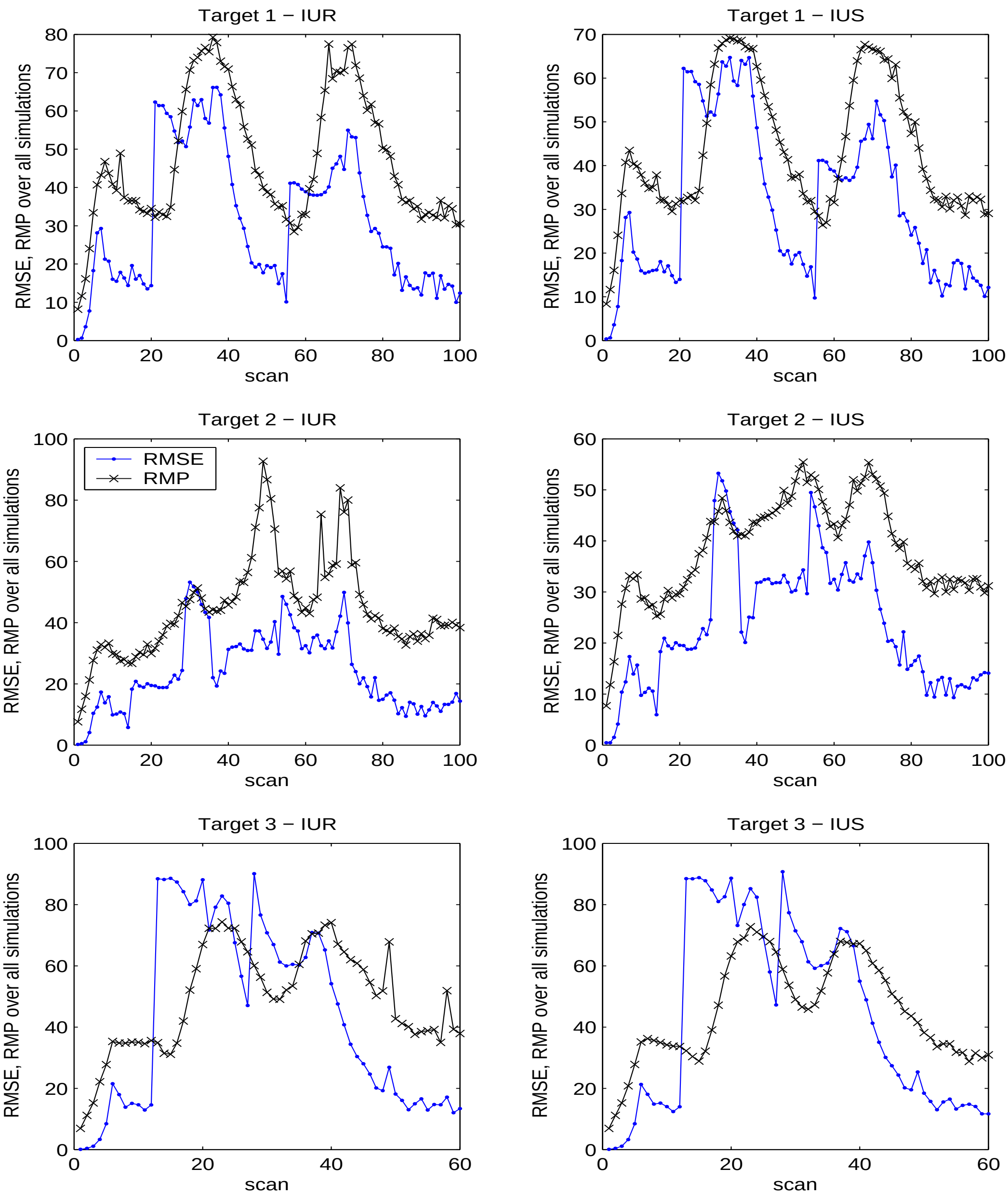

Figure 3.13: Targets 1 to 3 - Comparison of RMSE and RMP in acceleration estimation.

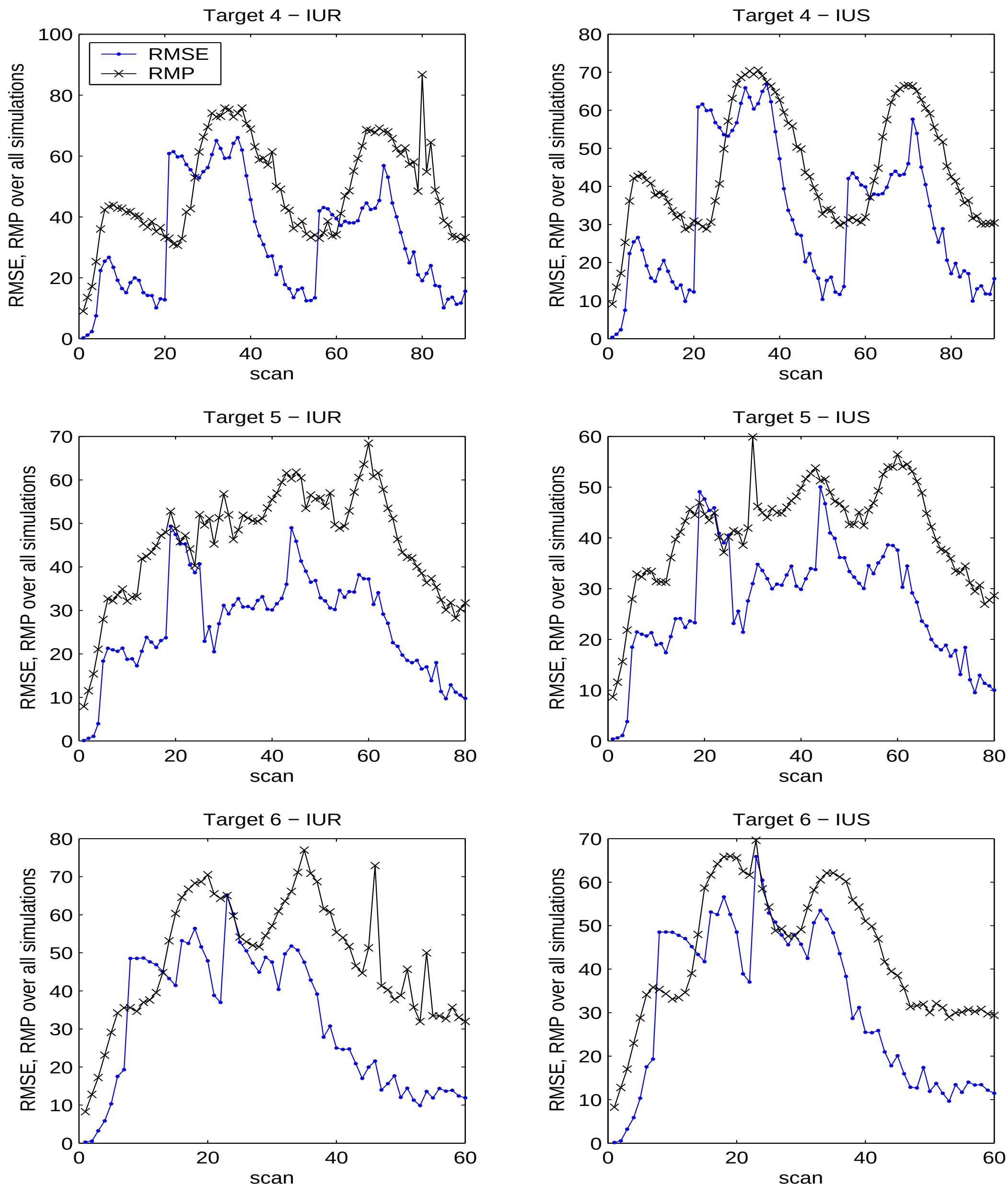

Figure 3.14: Targets 4 to 6 - Comparison of RMSE and RMP in acceleration estimation.

3.4.2 Target Tracking Using a Time Difference of Arrival System

Consider a problem on using a TDOA system to locate a target by processing measurements at four receiver stations. The measurements are then sent to a master station for computation of the time difference and the position estimates. In this section, we are concerned with 2D manœuvring target tracking.

Besides the IMM algorithm variants listed in Table 3.1, extended Kalman filter and unscented Kalman filter are also implemented for the simulation tests in this problem. EKF is used as the basis case for comparison of the filtering algorithms.

The target dynamics are modelled by the system represented by the process equation (Equation 3.39)

$$X_{k+1} = f(X_k, t_k, m_k) + g(X_k, t_k, m_k)w_k,$$

and the measurement/observation equation (Equation 3.40)

$$Z_{k+1} = h(X_{k+1}, t_{k+1}, m_{k+1}) + v_{k+1},$$

where $X_k = [x_k, y_k, \dot{x}_k, \dot{y}_k, \ddot{x}_k, \ddot{y}_k]^T$ is the state vector, and Z_k is the measurement vector.

At time step k, the state vector X_k is in mode $m_k \in \{1, 2, 3\}$, with m_k being the modal state of the system. The process noise vector w_k is zero-mean Gaussian-distributed with correlation matrix Q_j for model j, $j = 1, 2, 3$. Let I_n and 0_n denote the $n \times n$ identity and zero matrices respectively. The state evolutions within the different modes for each IMM algorithm variant are given as follows.

Model 1 (M_1 - CV model):

$$f(X_k, t_k, 1) = \begin{bmatrix} I_2 & TI_2 & 0_2 \\ 0_2 & I_2 & 0_2 \\ 0_2 & 0_2 & 0_2 \end{bmatrix} X_k, \quad g(X_k, t_k, 1) = \begin{bmatrix} 0.5T^2 I_2 \\ TI_2 \\ 0_2 \end{bmatrix}, \quad Q_1 = 5^2 I_2.$$

Model 2 (M_2 - CA model):

$$f(X_k, t_k, 2) = \begin{bmatrix} I_2 & TI_2 & 0.5T^2 I_2 \\ 0_2 & I_2 & TI_2 \\ 0_2 & 0_2 & I_2 \end{bmatrix} X_k, \quad g(X_k, t_k, 2) = \begin{bmatrix} 0.5T^2 I_2 \\ TI_2 \\ I_2 \end{bmatrix}, \quad Q_2 = 40^2 I_2.$$

Model 3 (M_3 - CT model):

$$f(X_k,t_k,3)=\begin{bmatrix}1 & 0 & \frac{\sin(\omega_k T)}{\omega_k} & -\frac{(1-\cos(\omega_k T))}{\omega_k} & 0 & 0\\ 0 & 1 & \frac{(1-\cos(\omega_k T))}{\omega_k} & \frac{\sin(\omega_k T)}{\omega_k} & 0 & 0\\ 0 & 0 & \cos(\omega_k T) & -\sin(\omega_k T) & 0 & 0\\ 0 & 0 & \sin(\omega_k T) & \cos(\omega_k T) & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix} X_k, g(X_k,t_k,3)=\begin{bmatrix}0.5T^2 I_2\\ T I_2\\ I_2\end{bmatrix},$$

$Q_3 = 7.5^2 I_2$, where $\omega_k = \dfrac{\dot{x}_k\ddot{y}_k - \dot{y}_k\ddot{x}_k}{\dot{x}_k^2+\dot{y}_k^2}$ is the turning rate.

Two cases are considered for the single filters, extended Kalman filter and unscented Kalman filter. In the first (respectively, second) case, the CA (respectively, CT) model for the IMM variants is used. The two cases are referred to as Case CA and Case CT respectively in this section.

Let the master station for the TDOA system be located at $[0,0]^T$, and the four receiver stations be located at

$$[\xi_i,\eta_i]^T = \begin{cases}[10000,10000]^T, & i=1,\\ [-10000,10000]^T, & i=2,\\ [-10000,-10000]^T, & i=3,\\ [10000,-10000]^T, & i=4.\end{cases}$$

Define $h_{ik} := \sqrt{x_k^2+y_k^2} + \sqrt{(x_k-\xi_i)^2+(y_k-\eta_i)^2}$, $i=1,\ldots,4$. The measurement function is mode-independent, given by

$$h(X_k,t_k,m_k) = \begin{bmatrix}h_{1k}, & h_{2k}, & h_{3k}, & h_{4k}\end{bmatrix}^T.$$

The measurement noise vector v_k is zero-mean Gaussian with covariance $R = 100^2 I_4$.

The Jacobian of the measurement equation required in the implementation of EKF is H, with the (i,j)-entry denoted by

$$H_{ij} := \begin{cases}\dfrac{x}{\sqrt{x^2+y^2}} + \dfrac{x-\xi_i}{\sqrt{(x-\xi_i)^2+(y-\eta_i)^2}}, & i=1,\ldots,4, j=1,\\ \dfrac{y}{\sqrt{x^2+y^2}} + \dfrac{y-\eta_i}{\sqrt{(x-\xi_i)^2+(y-\eta_i)^2}}, & i=1,\ldots,4, j=2,\\ 0, & i=1,\ldots,4, j=3,\ldots,6.\end{cases}$$

with evaluation done at the predicted state at each time step.

The transition probability matrix and the initial mode probability are

$$\begin{bmatrix} 0.96 & 0.02 & 0.02 \\ 0.02 & 0.96 & 0.02 \\ 0.02 & 0.02 & 0.96 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0.96 & 0.02 & 0.02 \end{bmatrix}$$

respectively. The sampling interval is $T = 1$ (second). For each particle filter used, the number of particles is $N_s = 300$. The total number of independent simulation runs is $C = 100$.

3.4.2.1 Scenario

Simulation tests are carried out on the target (see Figure 3.15) [58] described as follows. The initial state vector is $[21689, 10840, -\sqrt{80}, -\sqrt{159920}, 0, 0]^T$. From 0 to 20 seconds, it moves at constant velocity. From 20 to 35 seconds, it makes a coordinated turn to the right. From 35 to 55 seconds, it moves at constant velocity. From 55 to 70 seconds, it makes a coordinated turn to the left. From 70 to 87 seconds, it moves at constant velocity. The total number of scans during the tracking process is $L = 88$.

3.4.2.2 Computational Complexity

Let X be the notation for an arbitrary filtering algorithm among the ones implemented. The computational complexity of X is studied in the same way as it was done in Section 3.4.1.2.

1. processing time (in seconds) and analytic time complexity;
2. relationship between processing time and analytic time complexity;
3. sensitivity of the computation time required by a filtering algorithm with respect to the sampling interval.

1. Processing time and analytic time complexity

Let t_X (respectively, $\tilde{t}_X$) denote the mean processing time per simulation run (respectively, per scan) required by algorithm X. To measure the relative computational complexity of X with respect to EKF, the ratio t_X/t_{EKF} (respectively, $\tilde{t}_X/\tilde{t}_{EKF}$) is computed. Next, as done in Section 3.4.1.2, consider the number of operations required for one cycle of each filter and express it using the big-O notation (in terms of $r = 3$, $n_x = 6$ and $N_s = 300$). Tables 3.11 and 3.12 shows the computational complexity of each filtering algorithm.

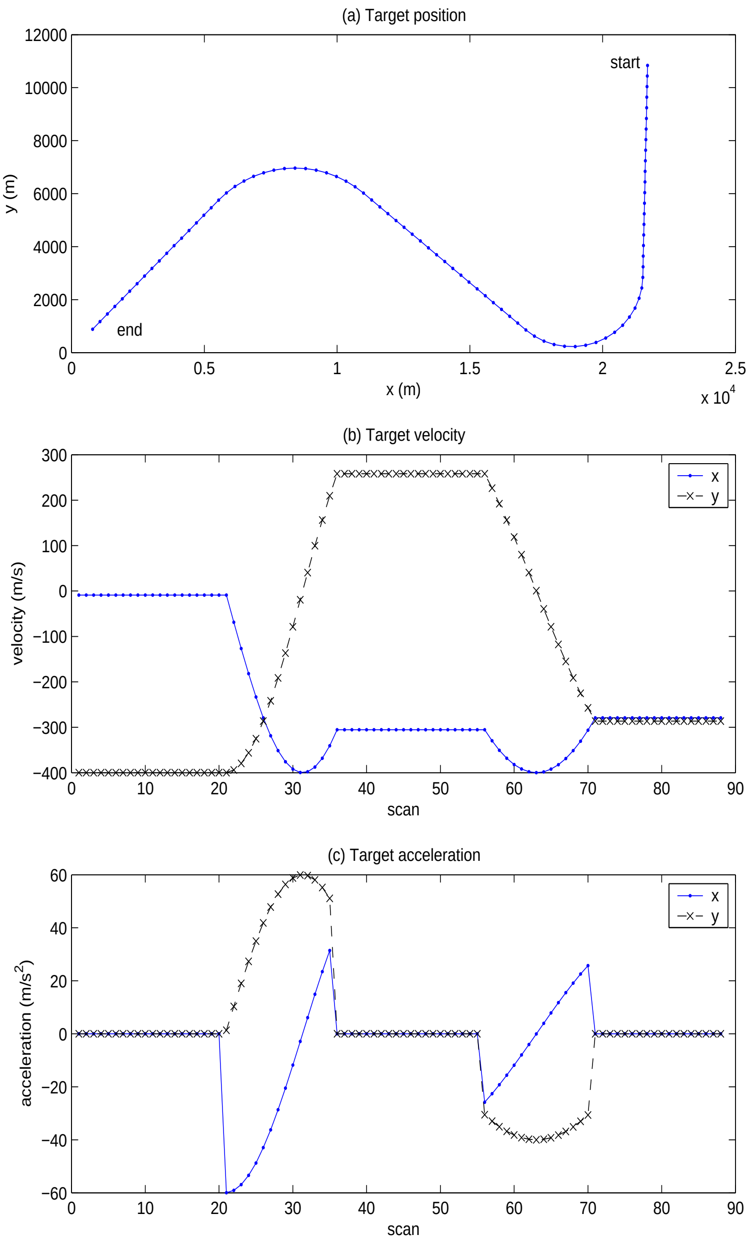

Figure 3.15: Trajectory of target manœuvring in 2D plane.

2. Relationship between processing time and analytic time complexity

Consider the relationship between the mean processing time per simulation run and the analytic time complexity for algorithm X. Let $O(b_X)$ denote the analytic time order for algorithm X, where $b(\cdot)$ is a function of r, n_x and N_s. Define the ratio

$$\hat{r}_X := \frac{\tau_X}{b_X}, \tag{3.54}$$

and the corresponding normalization with respect to $\hat{r}_{EKF}$,

$$r_X := \frac{\hat{r}_X}{\hat{r}_{EKF}}. \tag{3.55}$$

Filter X	Big-O notation	Mean processing time per run (s)	Mean processing time per scan (s)	Ratio with processing time for EKF
EKF	n_x^3	0.039	4e-04	1.00
UKF	n_x^3	0.065	0.001	1.65
IEK	$3n_x^3$	0.156	0.002	3.98
IUK	$3n_x^3$	0.215	0.002	5.51
IEE	$N_s n_x^3$	17.569	0.200	449.74
IEG	$N_s n_x^2$	8.976	0.102	229.78
IER	$N_s n_x^2$	5.484	0.062	140.39
IES	$N_s n_x^2$	5.472	0.062	140.09
IEU	$N_s n_x^3$	31.649	0.360	810.17
IEA	$N_s n_x^2$	12.302	0.140	314.92
IUE	$N_s n_x^3$	16.610	0.189	425.19
IUG	$N_s n_x^2$	8.638	0.098	221.13
IUR	$N_s n_x^2$	5.673	0.064	145.22
IUS	$N_s n_x^2$	5.655	0.064	144.76
IUU	$N_s n_x^3$	31.672	0.360	810.77
IUA	$N_s n_x^2$	12.239	0.139	313.29
IEUE	$N_s n_x^3$	16.575	0.188	424.31
IEUG	$N_s n_x^2$	8.675	0.099	222.06
IEUR	$N_s n_x^2$	5.674	0.064	145.26
IEUS	$N_s n_x^2$	5.659	0.064	144.86
IEUU	$N_s n_x^3$	31.823	0.362	814.63
IEUA	$N_s n_x^2$	12.221	0.139	312.84

Table 3.11: Case CA - Computational complexity.

From Figures 3.16 and 3.17, it can be seen that for each filtering algorithm X implemented, r_X is a positive constant of moderate magnitude. The results infer that the processing time for each filtering algorithm is proportional to the corresponding analytic time order. In addition, as mentioned in Section 3.4.1.2, the relative processing time for any two arbitrary filtering algorithms is proportional to their relative computational complexity (in terms of analytic time order).

3. Sensitivity of the computation time with respect to the sampling interval

Consider the sensitivity of the computation time required by a filtering algorithm with respect to the sampling interval $T = 1$ (second). The mean processing time per scan required by each filtering algorithm ranges from less than 0.001 seconds for EKF and UKF to less than 0.4 seconds for algorithms that use unscented particle filters, which is less than the sampling interval. Thus, for each algorithm implemented, there would be time for reaction to the target motion before the next scan.

Filter X	Big-O notation	Mean processing time per run (s)	Mean processing time per scan (s)	Ratio with processing time for EKF
EKF	n_x^3	0.044	5e-04	1.00
UKF	n_x^3	0.083	0.001	1.90
IEK	$3n_x^3$	0.155	0.002	3.52
IUK	$3n_x^3$	0.223	0.003	5.07
IEE	$N_s n_x^3$	18.619	0.212	424.43
IEG	$N_s n_x^2$	8.916	0.101	203.25
IER	$N_s n_x^2$	5.489	0.062	125.12
IES	$N_s n_x^2$	5.481	0.062	124.95
IEU	$N_s n_x^3$	34.188	0.388	779.31
IEA	$N_s n_x^2$	12.374	0.141	282.07
IUE	$N_s n_x^3$	17.758	0.202	404.80
IUG	$N_s n_x^2$	8.540	0.097	194.67
IUR	$N_s n_x^2$	5.666	0.064	129.16
IUS	$N_s n_x^2$	5.651	0.064	128.82
IUU	$N_s n_x^3$	34.207	0.389	779.75
IUA	$N_s n_x^2$	12.231	0.139	278.81
IEUE	$N_s n_x^3$	17.723	0.201	404.00
IEUG	$N_s n_x^2$	8.586	0.098	195.72
IEUR	$N_s n_x^2$	5.670	0.064	129.25
IEUS	$N_s n_x^2$	5.657	0.064	128.94
IEUU	$N_s n_x^3$	34.194	0.389	779.46
IEUA	$N_s n_x^2$	12.214	0.139	278.42

Table 3.12: Case CT - Computational complexity.

3.4.2.3 Analysis of Numerical Results

The approach for analysis of the simulation test results is the same as that in Section 3.4.1.3:

1. comparison of state estimation errors obtained with the different filtering algorithms;
2. statistical comparison of the filtering algorithms, formulated as a hypothesis problem;
3. consistency of the filtering algorithms.

For the discussion in this section, algorithms with occurrence(s) of divergence are omitted. UKF is denoted by UKF_CA and UKF_CT in Case CA and Case CT respectively.

1. Comparison of state estimation errors

Tables 3.13 and 3.14 show the root mean square errors (as defined by Equation 3.44) in state (namely, position, velocity and acceleration) estimation for the filtering algorithms implemented. For each algorithm X, $(\varepsilon_p)_X$ (respectively, $(\varepsilon_v)_X$, $(\varepsilon_a)_X$) denotes the RMSE in position (respectively, velocity, acceleration) estimation. The ratio

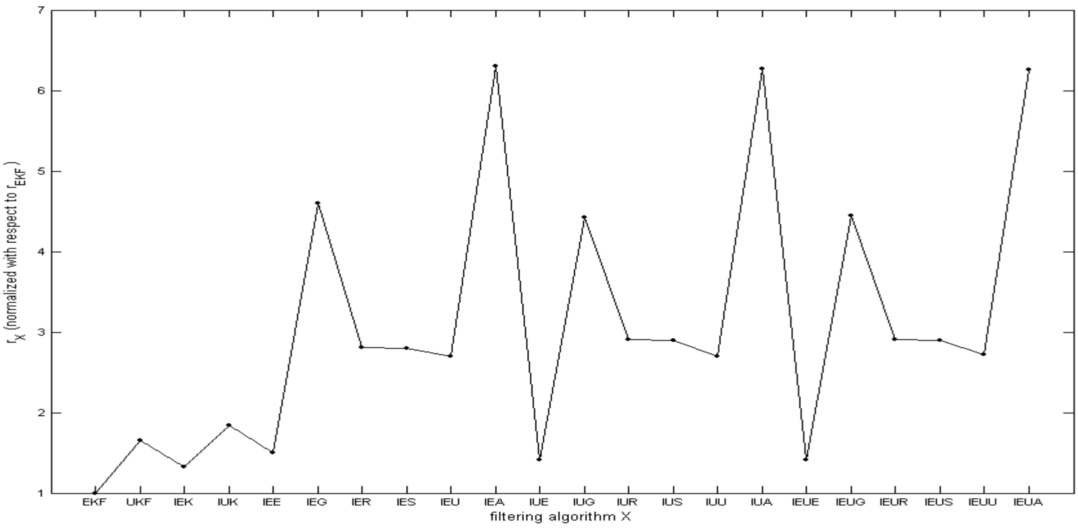

Figure 3.16: Case CA - Processing time relative to analytic time complexity.

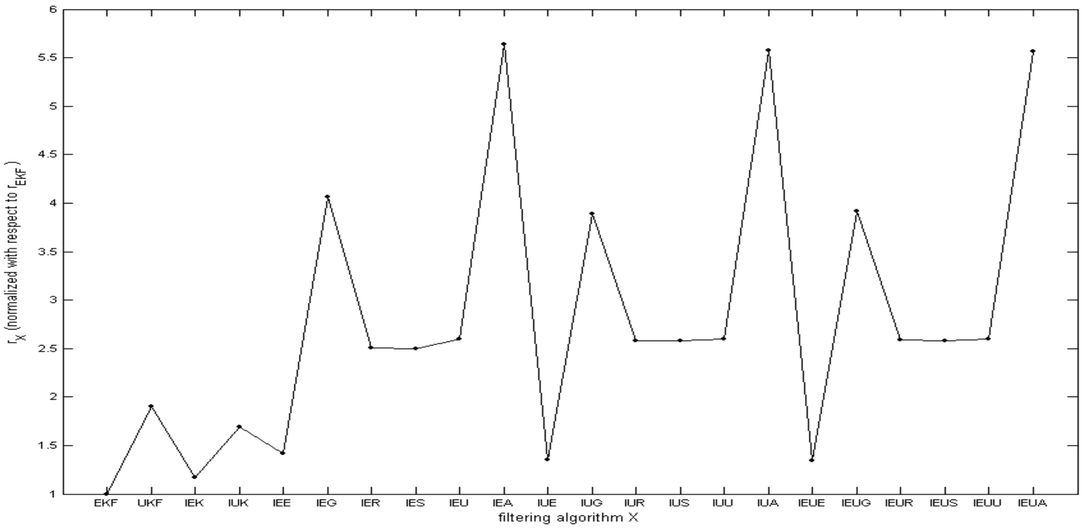

Figure 3.17: Case CT - Processing time relative to analytic time complexity.

$(\varepsilon_p)_X/(\varepsilon_p)_{EKF}$ (respectively, $(\varepsilon_v)_X/(\varepsilon_v)_{EKF}$, $(\varepsilon_a)_X/(\varepsilon_a)_{EKF}$) is computed as a measure of the relative RMSE in position (respectively, velocity, acceleration) estimation for algorithm X with respect to EKF. The notation "$O(10^n)$" (with n being a relevant positive integer) is used to represent very large errors in state estimation.

From the tables, it can be seen that state estimation errors obtained with EKF are significantly (up to a few hundred times) larger than the other filtering algorithms listed. The magnitudes of the state estimation errors obtained with the listed algorithms are summarized below.

Case CA

- position - EKF: $(\varepsilon_p)_{EKF}$ is $O(10^4)$;

- UKF: about 3% of $(\varepsilon_p)_{EKF}$;
- IUK, IUR, IUS, IUA, IEUR, IEUS, IEUA: about 0.8% of $(\varepsilon_p)_{EKF}$;

- velocity - EKF: $(\varepsilon_v)_{EKF}$ is $O(10^4)$;
  - UKF: about 2% of $(\varepsilon_v)_{EKF}$;
  - IUK, IUR, IUS, IUA, IEUR, IEUS, IEUA: about 0.5% of $(\varepsilon_v)_{EKF}$;

- acceleration - EKF: $(\varepsilon_a)_{EKF}$ is about 370;
  - UKF, IUK: about 12% of $(\varepsilon_a)_{EKF}$;
  - IUR, IUS, IUA, IEUR, IEUS, IEUA: about 8% of $(\varepsilon_a)_{EKF}$.

Case CT

- position - EKF: $(\varepsilon_p)_{EKF}$ is $O(10^3)$;
  - UKF: about 32% of $(\varepsilon_p)_{EKF}$;
  - IUK, IUR, IUS, IUA: about 7% of $(\varepsilon_p)_{EKF}$;
  - IEUR, IEUS, IEUA: about 4% of $(\varepsilon_p)_{EKF}$;

- velocity - EKF: $(\varepsilon_v)_{EKF} \approx 505$;
  - UKF: about 88% of $(\varepsilon_v)_{EKF}$;
  - IUK, IUR, IUS, IUA: about 33% of $(\varepsilon_v)_{EKF}$;
  - IEUR, IEUS, IEUA: about 12% of $(\varepsilon_v)_{EKF}$;

- acceleration - EKF: $(\varepsilon_a)_{EKF} \approx 79$;
  - UKF: about 87% of $(\varepsilon_a)_{EKF}$;
  - IUK: about 55% of $(\varepsilon_a)_{EKF}$;
  - IUK, IUR, IUS, IUA: about 32% of $(\varepsilon_a)_{EKF}$;
  - IEUR, IEUS, IEUA: about 36% of $(\varepsilon_a)_{EKF}$.

The preceding state estimation errors are interpreted as follows. Firstly, recall that EKF and UKF each provides Gaussian approximation to the posterior pdf of the system state. In addition, each particle in the framework of EKPF (respectively, UPF) uses an EKF (respectively, a UKF) to obtain a Gaussian approximation to the optimal proposal distribution. The objective is to attain more overlap between the proposal distribution and the posterior pdf, which has a Gaussian assumption imposed on its form. On the other hand, GPF approximates the posterior pdf by a single Gaussian distribution. In the current problem, the process model, as well as the measurement model, has possibly high nonlinearity. The actual posterior pdf in the state space may also be non-Gaussian. Consequently, it would be unlikely for EKF, UKF and the IMM algorithm variants

Filter X	RMSE in state estimation (to 2 decimal places)					
	position		velocity		acceleration	
	$(\varepsilon_p)_X$	$(\varepsilon_p)_X/(\varepsilon_p)_{EKF}$	$(\varepsilon_v)_X$	$(\varepsilon_v)_X/(\varepsilon_v)_{EKF}$	$(\varepsilon_a)_X$	$(\varepsilon_a)_X/(\varepsilon_a)_{EKF}$
EKF	$O(10^4)$	1.00	$O(10^4)$	1.00	368.53	1.00
UKF	400.79	0.03	225.33	0.02	38.34	0.10
IUK	89.56	6.73e-03	43.76	3.22e-03	43.50	0.12
IUR	92.43	6.95e-03	53.19	3.92e-03	24.85	0.07
IUS	92.51	6.96e-03	53.19	3.92e-03	24.85	0.07
IUA	92.64	6.97e-03	53.18	3.92e-03	24.84	0.07
IEUR	104.47	7.85e-03	59.18	4.36e-03	27.99	0.08
IEUS	104.66	7.87e-03	59.43	4.38e-03	28.11	0.08
IEUA	104.44	7.85e-03	59.36	4.37e-03	28.09	0.08

Table 3.13: Case CA - Errors in state estimation.

Filter X	RMSE in state estimation (to 2 decimal places)					
	position		velocity		acceleration	
	$(\varepsilon_p)_X$	$(\varepsilon_p)_X/(\varepsilon_p)_{EKF}$	$(\varepsilon_v)_X$	$(\varepsilon_v)_X/(\varepsilon_v)_{EKF}$	$(\varepsilon_a)_X$	$(\varepsilon_a)_X/(\varepsilon_a)_{EKF}$
EKF	$O(10^3)$	1.00	504.66	1.00	78.36	1.00
UKF	844.18	0.32	442.95	0.88	67.80	0.87
IUK	188.78	0.07	162.34	0.32	43.43	0.55
IUR	187.14	0.07	162.50	0.32	24.99	0.32
IUS	186.83	0.07	162.34	0.32	24.98	0.32
IUA	190.80	0.07	165.80	0.33	24.99	0.32
IEUR	105.06	0.04	60.83	0.12	28.41	0.36
IEUS	105.24	0.04	61.13	0.12	28.39	0.36
IEUA	105.14	0.04	60.60	0.12	28.26	0.36

Table 3.14: Case CT - Errors in state estimation.

which use EKPF, UPF or GPF to have good performance for this problem. Secondly, large state estimation errors or divergence for filters could be due to the accumulation of rounding errors brought about by the large number of mathematical operations required for the current problem, especially when particle filters are implemented. Thirdly, it is also possible that a larger number of particles N_s is necessary for stability to be attained. However, this implies that implementing particle filters (especially the computationally intensive ones) for the IMM algorithm variants would probably incur prohibitive costs when solving larger problems, such as those with n_x or N_s increased.

It is also noted that the results in position and velocity estimation for IMM algorithm variants are comparable, but IUK requires a much lower (at least 25 times less) computational cost than the other three algorithms which use particle filters. UKF has an even lower computational load of about one-third that of IUK, but the state estimation errors obtained are much larger.

2. Statistical comparison of filtering algorithms

As in Section 3.4.1.3, comparison of the performance (in terms of the mean square error in state estimation) of the filtering algorithms implemented is formulated as a hypothesis testing problem as follows [13, Sections 1.5 and 11.5].

Let X be an arbitrary filtering algorithm implemented in the preceding simulation tests. Let J_X denote the actual mean square error in state estimation. The hypothesis testing problem is to test the null hypothesis

$$H_0 : \Delta = J_{EKF} - J_X \leq 0 \quad \text{(algorithm } X \text{ not better than EKF)}, \tag{3.56}$$

versus the alternate hypothesis

$$H_1 : \Delta = J_{EKF} - J_X > 0 \quad \text{(algorithm } X \text{ better than EKF)}, \tag{3.57}$$

subject to

$$\text{Prob}\{\text{accept } H_1 | H_0 \text{ is true}\} = \alpha \quad \text{(level of significance for hypothesis } H_0\text{)}.$$

With reference to Equation 3.48, the statistical test is based on the mean square error differences,

$$(\Delta_X)_i := \frac{1}{L}\sum_{k=1}^{L}\Big\{\big\|(\hat{u}_{EKF})_{ik} - u_k\big\|_2^2 - \big\|(\hat{u}_X)_{ik} - u_k\big\|_2^2\Big\}, \quad i = 1, \ldots, C, \tag{3.58}$$

which are independent from run to run. The test uses the sample mean of the above differences (Equation 3.49),

$$\bar{\Delta}_X := \frac{1}{C}\sum_{i=1}^{C}(\Delta_X)_i,$$

and its associated standard deviation (Equation 3.50),

$$\sigma_{\bar{\Delta}_X} := \Big\{\frac{1}{C^2}\sum_{i=1}^{C}\big((\Delta_X)_i - \bar{\Delta}_X\big)^2\Big\}^{\frac{1}{2}}.$$

As in Section 3.4.1.3, we use $\alpha = 0.05$ (equivalently, 5% level of significance). Thus, H_1 is accepted if the test statistic

$$\rho = \frac{\bar{\Delta}_X}{\sigma_{\bar{\Delta}_X}} > 1.65.$$

Tables 3.15 and 3.16 show the test for differences of mean square errors in state estimation between EKF and each of the other filtering algorithms.

It can be observed that UKF_CA and all the IMM algorithm variants listed in the tables have statistically significant improvement over EKF (that is, test statistic $\rho > 1.65$)

Filter X	Test statistic for state estimation		
	position	velocity	acceleration
UKF	11.65	11.59	12.66
IUK	11.68	11.59	12.67
IUR	11.68	11.59	12.79
IUS	11.68	11.59	12.79
IUA	11.68	11.59	12.79
IEUR	11.68	11.59	12.77
IEUS	11.68	11.59	12.77
IEUA	11.68	11.59	12.77

Table 3.15: Case CA - Comparison of EKF with other filters in state estimation.

Filter X	Test statistic for state estimation		
	position	velocity	acceleration
UKF	7.86	0.30	3.77
IUK	19.63	7.96	11.60
IUR	19.64	8.20	15.09
IUS	19.64	8.22	15.09
IUA	19.63	7.90	15.09
IEUR	19.90	18.74	14.58
IEUS	19.90	18.73	14.56
IEUA	19.90	18.73	14.63

Table 3.16: Case CT - Comparison of EKF with other filters in state estimation.

in position, velocity and acceleration estimation. UKF_CT also has statistically significant improvement over EKF in position and acceleration estimation. However, the improvement in velocity estimation with UKF_CT is not significant enough to reject the null hypothesis H_0 at 5% level of significance.

Based on the statistical comparison of the filtering algorithms, it can be inferred that, in terms of state estimation errors, the IMM algorithms listed in Tables 3.15 and 3.16 have better performance than EKF.

3. Consistency of filtering algorithms

The consistency of a filtering algorithm X is determined by comparing the state estimation errors obtained with X and the corresponding filter-calculated state error covariances.

As in Section 3.4.1.3, let $(\hat{u}_X)_{ik}$ and $(\hat{P}_X)_{ik}$ be the state estimate and the corresponding state error covariance for algorithm X in the i-th simulation run, $i = 1, \ldots, C$, and u_k be the actual state, $k = 1, \ldots, L$. A comparison of state estimation errors defined in

Equation 3.52,

$$\mathrm{RMSE}_X(k) := \left\{\frac{1}{C}\sum_{i=1}^{C}\left\|(\hat{u}_X)_{ik} - u_k\right\|_2^2\right\}^{\frac{1}{2}},$$

and filter-calculated state error covariances represented by Equation 3.53,

$$\mathrm{RMP}_X(k) := \left\{\frac{1}{C}\sum_{i=1}^{C}\mathrm{trace}((\hat{P}_X)_{ik})\right\}^{\frac{1}{2}},$$

is done for the filtering algorithms listed in Tables 3.15 and 3.16. The state estimation errors (label used: RMSE) and filter-calculated state error covariances (label used: RMP) are shown in Figures 3.18 to 3.23. The observations for each state are described below.

- position (Figures 3.18 and 3.19):

  For the IMM algorithm variants, RMSE is almost commensurate with RMP during non-manœuvring periods. During periods of manœuvres, RMSE is smaller than RMP. For UKF_CA, RMSE is larger than RMP at the initial stage of tracking but subsequently remains smaller than RMP. For UKF_CT, RMSE is larger than RMP at the initial stage of tracking. Subsequently (after the onset of coordinated turn motion), RMSE becomes almost identical to RMP. The reason could be that the filter matches the model of the actual system.

- velocity (Figures 3.20 and 3.21):

  For IUK, IUR, IUS and IUA, RMSE follows the trend of RMP, but RMSE is generally smaller than RMP, except at the start of tracking and possibly at the beginning of a manœuvre. For UKF_CA, IEUR, IEUS and IEUA, RMSE is significantly smaller than RMP throughout tracking, except at the initial stage of tracking. For UKF_CT, RMSE is larger than RMP during at the initial stage of tracking and during manœuvres. At other times during tracking, RMSE is smaller than RMP.

- acceleration (Figures 3.22 and 3.23):

  For IUK, IUR, IUS and IUA, RMSE and RMP have the same behaviour during tracking, but RMSE is generally smaller than RMP, except at the onset of or during a manœuvre. For UKF_CA, IEUR, IEUS and IEUA, RMSE is significantly smaller than RMP throughout tracking, except at the initial stage of tracking. For UKF_CT, RMSE is smaller than RMP most of the time before scan 20, when a manœuvre begins. After scan 28, RMSE remains larger than RMP.

Based on the first two parts on analysis of the numerical results, IUK, IUR, IUS, IUA, IEUR, IEUS and IEUA have better overall performance than EKF, in terms of smaller state estimation errors obtained. But they are not totally consistent because not all the state estimation errors are commensurate with the filter-calculated state error covariances. The situation for IEUR, IEUS and IEUA is worse than IUR, IUS and IUA respectively. This is shown by the larger discrepancies in the trends of velocity and acceleration estimation errors, as well as the large differences in the magnitudes of state estimation errors. It was mentioned earlier that this is not an unusual phenomenon for adaptive algorithms [12], with timely adaptation being driven by the inconsistency. It is of interest in practice to strive to tune the filters to improve their consistency [13].

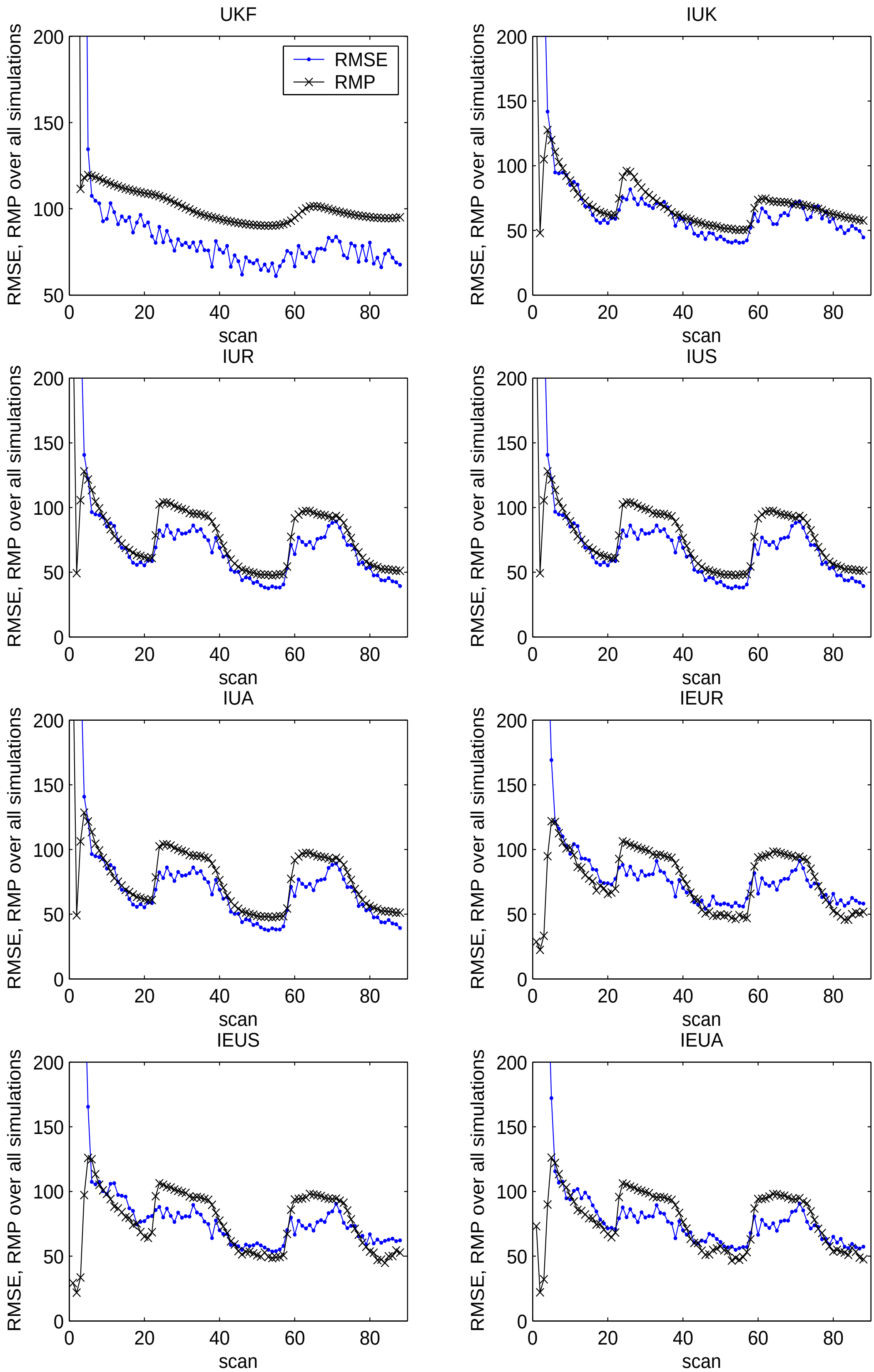

Figure 3.18: Case CA - Comparison of RMSE and RMP in position estimation.

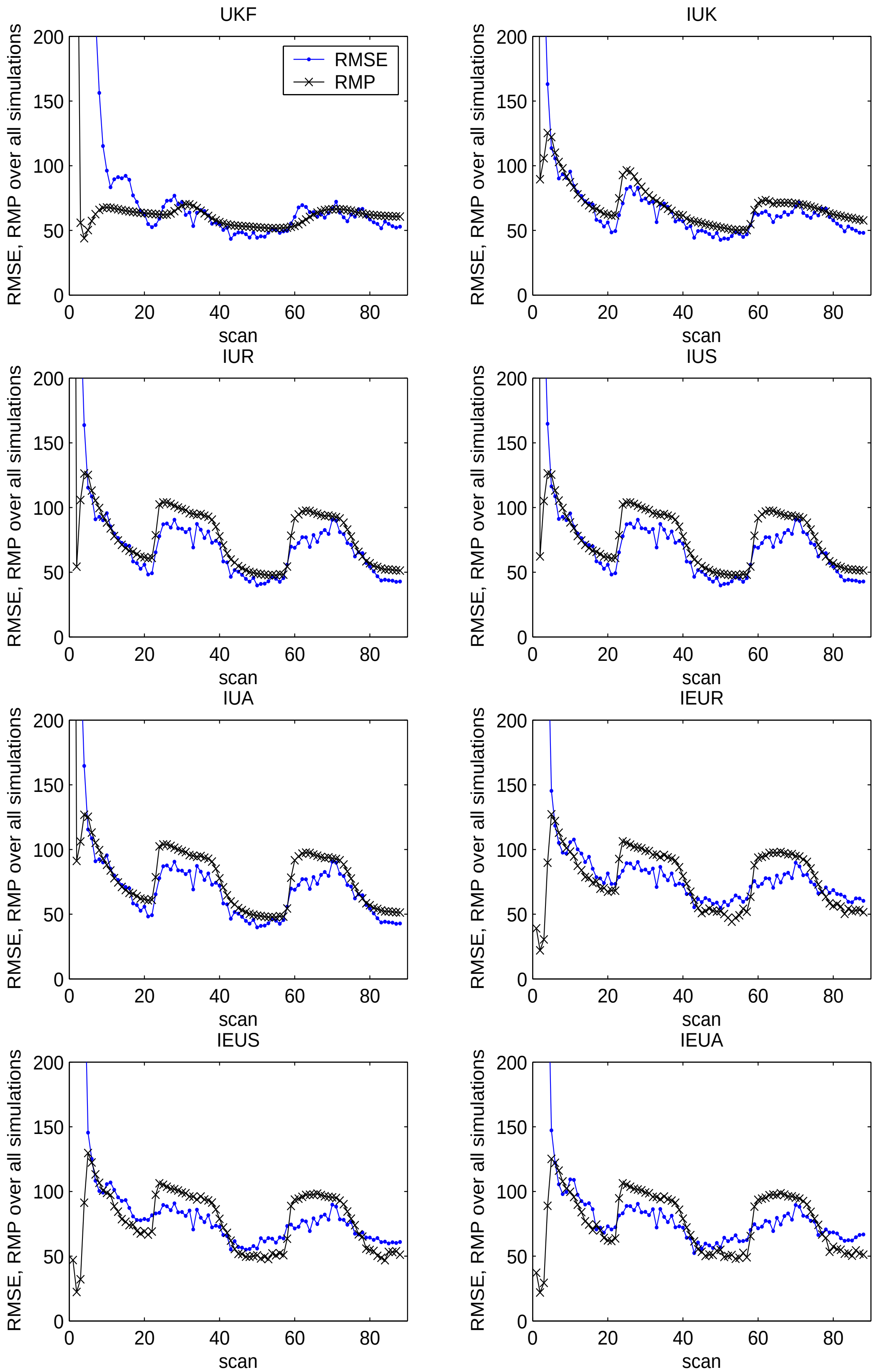

Figure 3.19: Case CT - Comparison of RMSE and RMP in position estimation.

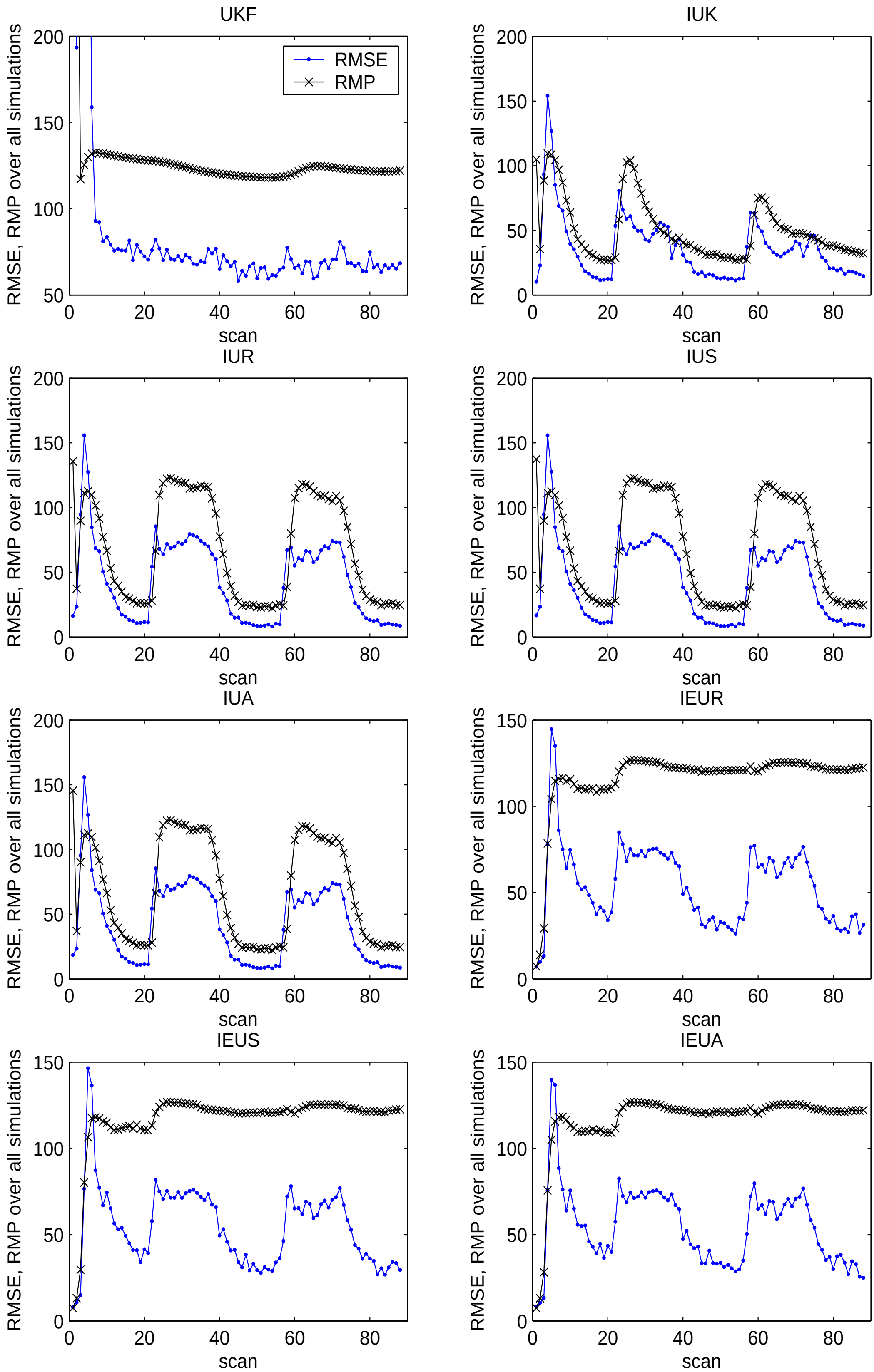

Figure 3.20: Case CA - Comparison of RMSE and RMP in velocity estimation.

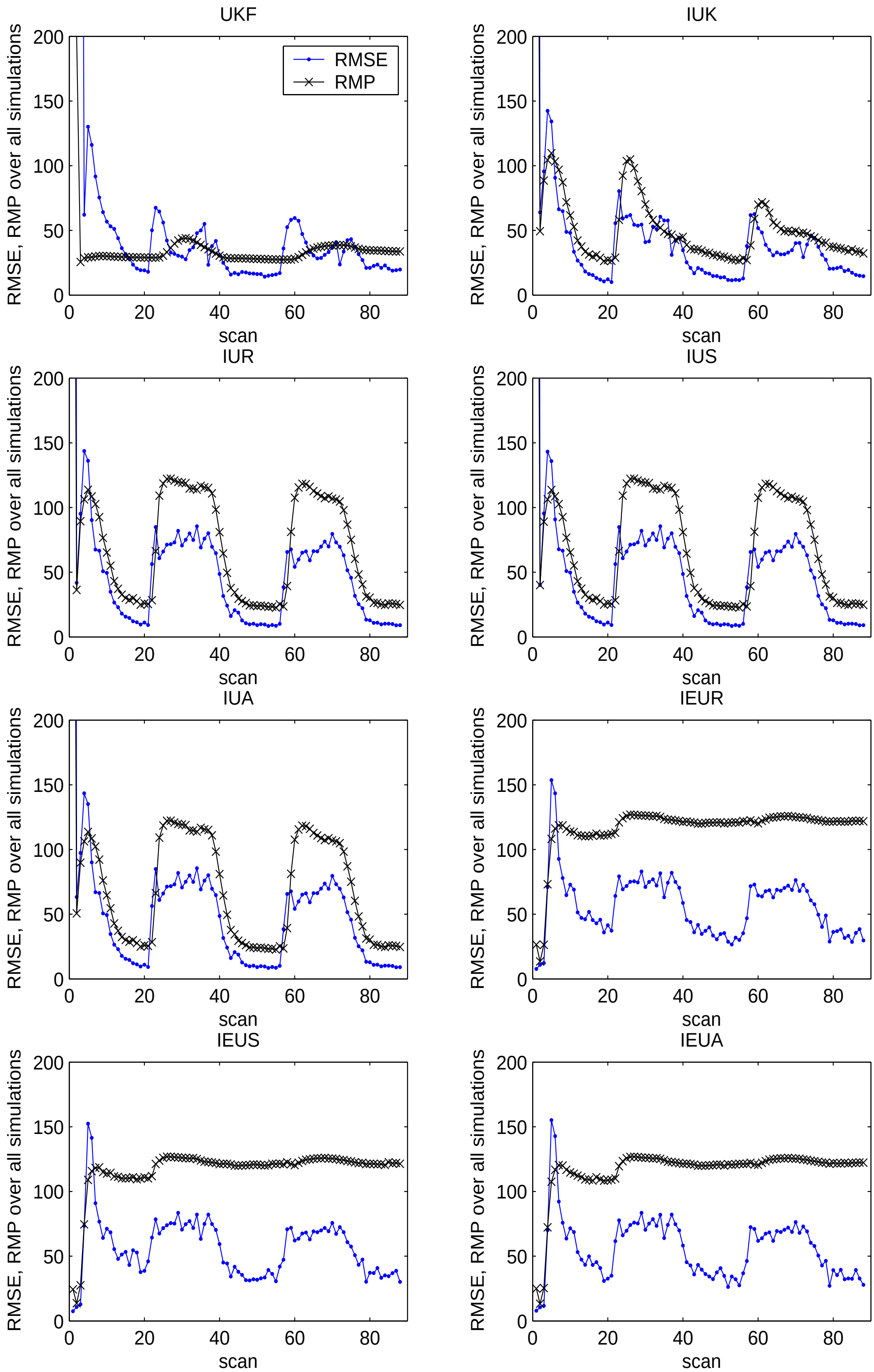

Figure 3.21: Case CT - Comparison of RMSE and RMP in velocity estimation.

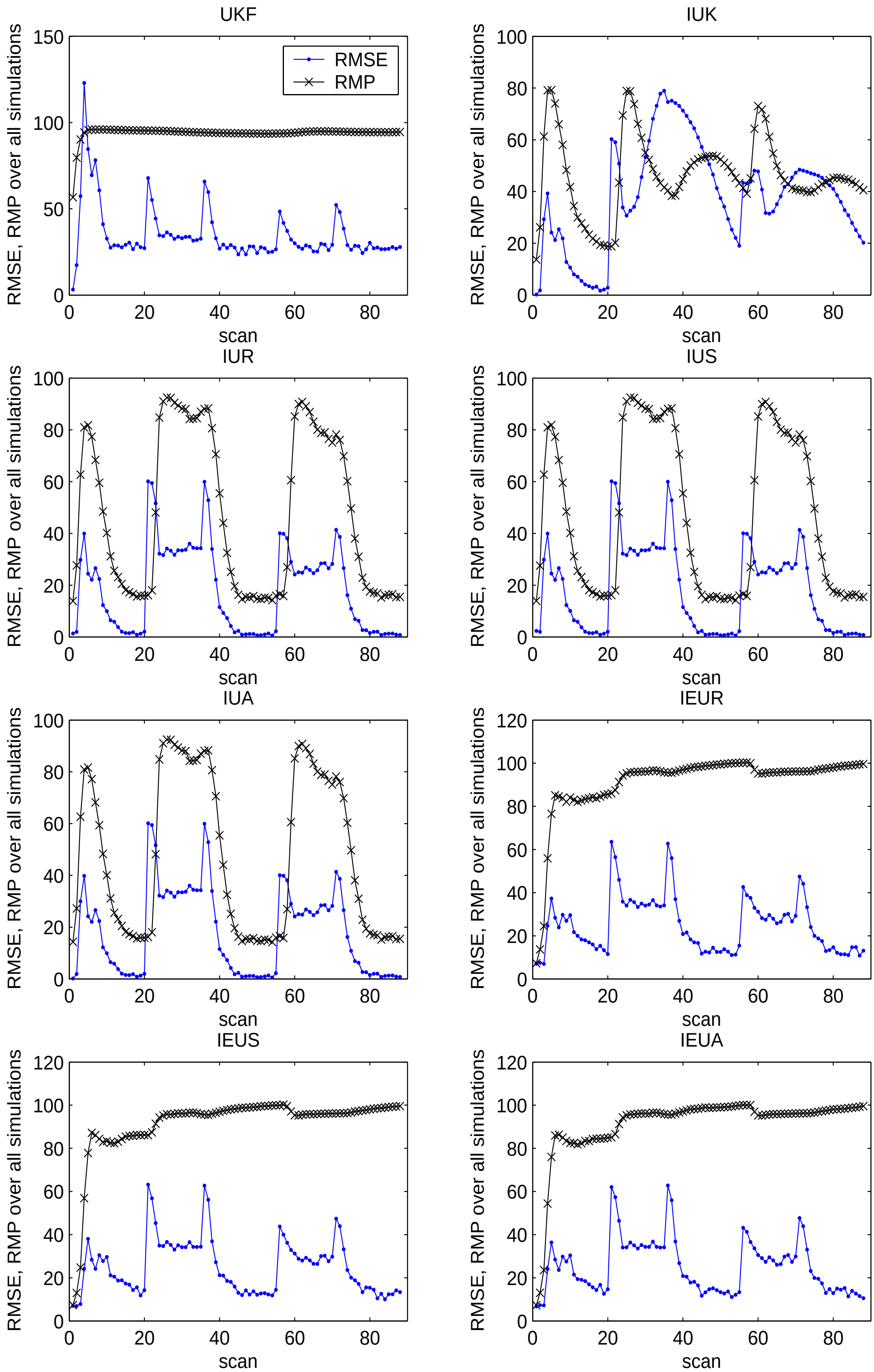

Figure 3.22: Case CA - Comparison of RMSE and RMP in acceleration estimation.

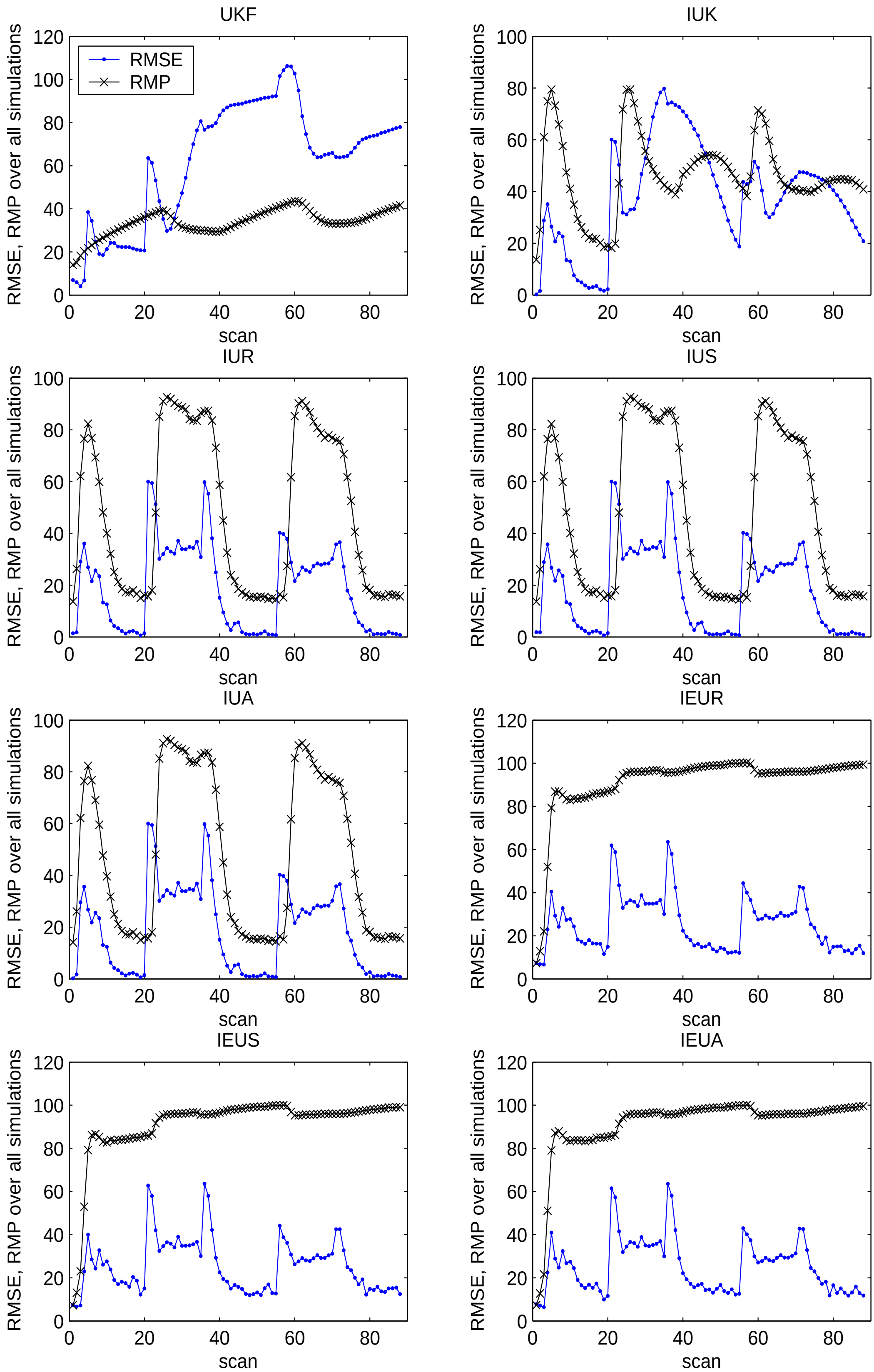

Figure 3.23: Case CT - Comparison of RMSE and RMP in acceleration estimation.

3.5 Application: Modelling Financial Option Prices

A *derivative* is a financial instrument whose value depends on (or derives from) the values of other more basic underlying variables, which are often prices of traded assets [146]. Options are contracts that are traded actively in financial markets. An *option* is a derivative that gives the holder the right to do something. A *call option* (respectively, *put option*) gives the holder the right to buy (respectively, sell) the underlying asset by a specified date in the future, for a predetermined price. The date in the contract is known as the *maturity* (or *expiration date*). The price in the contract is known as the *strike price* (or *exercise price*).

The Black-Scholes (or Black-Scholes-Merton) model is widely used to model the behaviour of stock prices [146]. The following assumptions are used.

1. The stock price S follows the process (geometric Brownian motion):

$$dS = \mu S dt + \sigma S dz, \tag{3.59}$$

   where t is time, μ is the expected rate of return on the stock, σ is the volatility of the stock price, $dz = \lim_{\Delta t \to 0} \epsilon\sqrt{\Delta t}$ and ϵ is a random sample drawn from the standard Gaussian distribution $\mathcal{N}(0, 1)$ (mean is 0 and standard deviation is 1), with μ and σ constant.

2. There are no transaction costs and no dividends.

3. There are no risk-less arbitrage opportunities.

4. An instantaneous risk-less portfolio is used.

5. Trading is continuous.

6. The risk-free interest rate λ is constant.

Let f be the price of an option that is dependent on a stock. Let t_m be the maturity of the option. The variable f needs to satisfy the Black-Scholes (or Black-Scholes-Merton) differential equation [146]

$$\frac{\partial f}{\partial t} + \lambda S \frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = \lambda f, \tag{3.60}$$

with relevant boundary conditions.

At a general time t, the Black-Scholes formulas for the price of a European option on a stock can be derived by solving Equation 3.60, subject to the key boundary condition [146]

- $f = \max(S - X, 0)$ when $t = t_m$, in the case of a call option,

- $f = \max(X - S, 0)$ when $t = t_m$, in the case of a put option.

The formulas are

$$c = SN(d_1) - Xe^{-\lambda\psi}N(d_2) \tag{3.61}$$

and

$$p = -SN(-d_1) + Xe^{-\lambda\psi}N(-d_2), \tag{3.62}$$

where c is the price of a call option, p is the price of a put option, S is the stock price at time t, X is the strike price, ψ is the time (in years) to maturity (that is, $\psi = t_m - t$),

$$\begin{aligned} d_1 &= \frac{\ln(S/X) + (\lambda + \sigma^2/2)\psi}{\sigma\sqrt{\psi}}, \\ d_2 &= \frac{\ln(S/X) + (\lambda - \sigma^2/2)\psi}{\sigma\sqrt{\psi}} = d_1 - \sigma\sqrt{\psi} \end{aligned} \tag{3.63}$$

and $N(\cdot)$ is the cumulative probability distribution function for a standard Gaussian distribution. A polynomial approximation for the cumulative standard Gaussian distribution $N(\cdot)$ is computed by

$$N(x) \approx \begin{cases} 1 - N'(x)(a_1\gamma + a_2\gamma^2 + a_3\gamma^3 + a_4\gamma^4 + a_5\gamma^5) & \text{when } x \geq 0, \\ 1 - N(-x) & \text{when } x < 0, \end{cases}$$

where

$$\begin{aligned} &N'(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}, && \gamma = \frac{1}{1 + \gamma_0 x}, && \gamma_0 = 0.2316419, \\ &a_1 = 0.319381530, && a_2 = -0.356563782, && a_3 = 1.781477937, \\ &a_4 = -1.821255978, && a_5 = 1.330274429. && \end{aligned}$$

The volatility is frequently estimated from historical data [146]. This can be done reasonably well by using daily data over the most recent 90 to 180 days. The risk-free interest rate can be estimated by monitoring interest rates in the bond markets.

3.5.1 Simulation Tests

The IMM algorithm variants listed in Table 3.1 and the following nonlinear filters discussed in Section 3.3, namely, EKF, UKF, SPF, APF, RPF, EKPF, UPF and GPF, are implemented. EKF is used as the basis case for comparison of the filtering algorithms. The approach to performance evaluation is similar to that in Section 3.4.

The numerical tests done here follow from [240,320]. Consider the state-space model of the system for recursive estimation of the option prices:

$$X_k = X_{k-1} + w_{k-1}, \quad k \in \mathbb{N}, \tag{3.64}$$

and the measurement/observation equation

$$Z_k = h(X_k, e_k) + v_k, \quad k \in \mathbb{N}. \tag{3.65}$$

The state vector consists of the interest rate λ and the volatility σ, while the measurement vector comprises the call option price c and the put option price p. The time to maturity ψ and the stock price S are used as input information. In symbol form, at time step k,

$$X_k = [\lambda_k, \sigma_k]^T, \quad Z_k = [c_k, p_k]^T, \quad e_k = [\psi_k, S_k]^T. \tag{3.66}$$

With reference to Section 3.2, $n_x = n_z = n_e = n_w = n_v = 2$. The covariances of the process noise and the measurement noise are assumed to be known and are set to small diagonal matrices.

With reference to Equations 3.1 and 3.2, $f(\cdot)$ and $g(\cdot)$ are both identity functions in this problem. The Jacobians of the process equation required in the implementation of EKF are I_{n_x} and $I_{n_x \times n_w}$ respectively. The Jacobian of the measurement equation is

$$H = \begin{bmatrix} \frac{\partial c}{\partial \lambda} & \frac{\partial c}{\partial \sigma} \\ \frac{\partial p}{\partial \lambda} & \frac{\partial p}{\partial \sigma} \end{bmatrix}, \tag{3.67}$$

where

$$\frac{\partial c}{\partial \lambda} = X\psi e^{-\lambda\psi} N(d_2), \quad \frac{\partial p}{\partial \lambda} = -X\psi e^{-\lambda\psi} N(-d_2), \quad \frac{\partial c}{\partial \sigma} = \frac{\partial p}{\partial \sigma} = S\sqrt{\psi} N'(d_1), \tag{3.68}$$

with evaluation done at the predicted state at each time step (see Section A.3 for the derivation of H).

For each IMM algorithm, the modal state of the system is taken into consideration at each time step. The transition probability matrix and the initial mode probability are the same as those in Section 3.4. For each particle filter implemented, the number of particles used is $N_s = 100$. The total number of independent simulation runs is $C = 100$.

The test data set used comprises five pairs of call and put option contracts on the British FTSE-100 index over the period February 1994 to December 1994 [240, 320]. The total number of trading days in the data set is 224. The option pairs have identical strike price $X \in \{2925, 3025, 3125, 3225, 3325\}$ and maturity t_m. A different volatility can be estimated for each option.

3.5.1.1 Computational Complexity

Firstly, consider the processing time (in seconds). Let t_A denote the mean processing time per simulation run required by algorithm A. Define τ_A as the average of t_A calculated over the five sets of simulation tests. To measure the relative computational complexity of A with respect to EKF, the ratio τ_A/τ_{EKF} is computed. Then as done in Sections 3.4.1.2 and 3.4.2.2, express the number of operations required for one cycle of each algorithm using the big-O notation (in terms of $r = 3$, $n_x = 2$ and $N_s = 100$).

Table 3.17 shows the computational complexity of each filtering algorithm. The last column displays the relative computational complexity of each algorithm with respect to EKF.

Filter A	Big-O notation	Mean processing time per simulation run (s)						
		2925	3025	3125	3225	3325	Average	τ_A/τ_{EKF}
EKF	n_x^3	0.10	0.11	0.11	0.10	0.11	0.11	1.00
UKF	n_x^3	0.14	0.15	0.14	0.14	0.14	0.14	1.34
EKPF	$N_s n_x^3$	9.41	9.77	9.73	9.76	9.75	9.69	92.06
GPF	$N_s n_x^2$	6.47	6.76	6.75	6.79	6.77	6.71	63.75
RPF	$N_s n_x^2$	4.22	4.46	4.45	4.48	4.45	4.41	41.94
SPF	$N_s n_x^2$	4.21	4.44	4.43	4.46	4.44	4.40	41.78
UPF	$N_s n_x^3$	13.68	14.04	14.03	14.05	14.02	13.96	132.73
APF	$N_s n_x^2$	9.51	9.88	9.87	9.91	9.89	9.81	93.24
IEK	$3n_x^3$	0.30	0.32	0.31	0.31	0.32	0.31	2.98
IUK	$3n_x^3$	0.41	0.43	0.42	0.42	0.43	0.42	3.99
IEE	$N_s n_x^3$	9.72	10.10	10.05	10.08	10.07	10.01	95.10
IEG	$N_s n_x^2$	6.84	7.20	7.21	7.19	7.22	7.13	67.78
IER	$N_s n_x^2$	4.56	4.80	4.80	4.82	4.81	4.76	45.23
IES	$N_s n_x^2$	4.56	4.80	4.80	4.82	4.80	4.76	45.20
IEU	$N_s n_x^3$	14.04	14.43	14.42	14.43	14.43	14.35	136.40
IEA	$N_s n_x^2$	9.86	10.25	10.24	10.28	10.26	10.18	96.74
IUE	$N_s n_x^3$	9.81	10.19	10.14	10.17	10.16	10.10	95.96
IUG	$N_s n_x^2$	6.94	7.27	7.28	7.30	7.26	7.21	68.52
IUR	$N_s n_x^2$	4.65	4.89	4.89	4.91	4.89	4.85	46.06
IUS	$N_s n_x^2$	4.65	4.89	4.88	4.90	4.89	4.84	46.01
IUU	$N_s n_x^3$	14.08	14.46	14.46	14.47	14.46	14.38	136.72
IUA	$N_s n_x^2$	9.93	10.33	10.32	10.36	10.34	10.26	97.48
IEUE	$N_s n_x^3$	9.79	10.18	10.13	10.15	10.15	10.08	95.82
IEUG	$N_s n_x^2$	6.94	7.26	7.24	7.29	7.25	7.20	68.39
IEUR	$N_s n_x^2$	4.64	4.88	4.88	4.89	4.88	4.83	45.94
IEUS	$N_s n_x^2$	4.63	4.88	4.87	4.89	4.88	4.83	45.92
IEUU	$N_s n_x^3$	14.08	14.46	14.45	14.47	14.46	14.38	136.72
IEUA	$N_s n_x^2$	9.93	10.33	10.32	10.35	10.34	10.25	97.45

Table 3.17: Computational complexity (per simulation run).

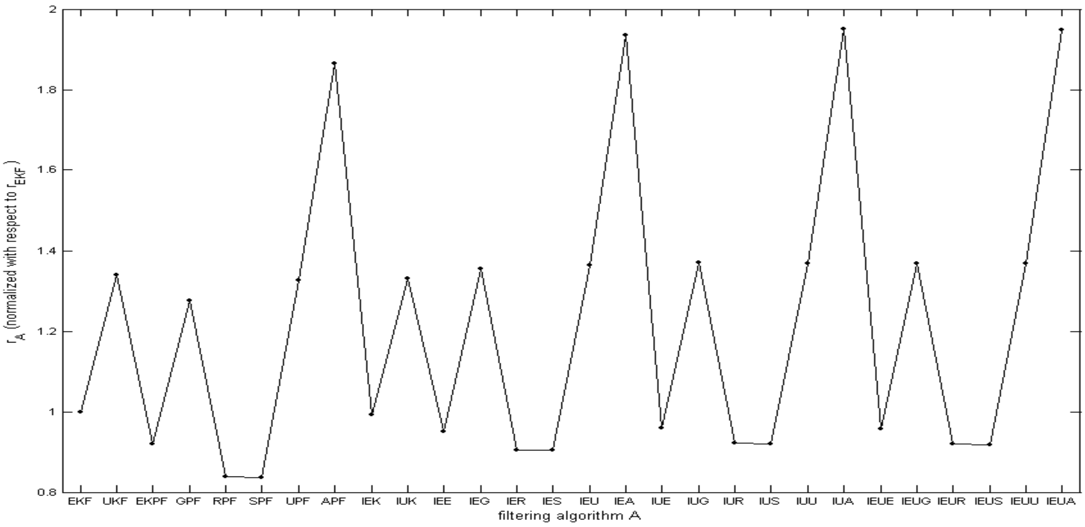

Figure 3.24: Processing time relative to analytic time complexity.

As done previously in Sections 3.4.1.2 and 3.4.2.2, consider the relationship between the mean processing time per simulation run and the analytic time complexity for each filtering algorithm A implemented. Let $O(b_A)$ denote the analytic time order for algorithm A, where $b(\cdot)$ is a function of r, n_x and N_s. We use the definitions from Equations 3.54 and 3.55 to get

$$\hat{r}_A := \frac{\tau_A}{b_A},$$

and the corresponding normalization with respect to $\hat{r}_{EKF}$,

$$r_A := \frac{\hat{r}_A}{\hat{r}_{EKF}}.$$

Figure 3.24 shows that for each filtering algorithm A implemented, the ratio r_A is a positive constant of moderate magnitude. The results indicate that the processing time for each filtering algorithm is proportional to its computational complexity (in terms of the corresponding analytic time order). As mentioned in Sections 3.4.1.2 and 3.4.2.2, the relative processing time for any two arbitrary filtering algorithms is proportional to their relative computational complexity.

3.5.1.2 Analysis of Numerical Results

Analysis of the simulation test results is carried out as follows:

1. comparison of state estimation errors obtained with the different filtering algorithms;
2. statistical comparison of the filtering algorithms, formulated as a hypothesis problem.

1. Comparison of estimation errors

Let L denote the total number of trading days in the data set. The root mean square errors in the estimation of call (respectively, put) option price is computed over the last 100 days of trading, to allow convergence of the algorithms. The errors are defined as

$$\text{RMSE} := \left\{ \frac{1}{C} \frac{1}{100} \sum_{i=1}^{C} \sum_{k=L-99}^{L} \left| \hat{u}_{ik} - u_k \right|_2^2 \right\}^{\frac{1}{2}},$$

where

- $\hat{u}_{ik}$ is the estimate of the call (respectively, put) option price at the k-th scan in the i-th simulation run, $i = 1, \ldots, C$, $k = 1, \ldots, L$, and
- u_k is the actual call (respectively, put) option price at the k-th scan, $k = 1, \ldots, L$.

Tables 3.18 and 3.19 show the RMSEs in the estimation of call option prices and put option prices respectively, for the filtering algorithm implemented (algorithms with occurrence(s) of divergence are omitted). For each algorithm, the average of RMSE in the estimation of option prices, calculated over the five call/put options, are also tabulated. In the tables, each option is represented by its strike price. The notation ϕ_A denotes the average RMSE in the estimation of call option (respectively, put option) price for algorithm A in Table 3.18 (respectively, Table 3.19).

It can be observed from the tabulated results that algorithms which use EKPF and UPF yield relatively smaller estimation errors than the other algorithms. They attain average RMSEs that are about 25% to 35% of those obtained with EKF. This is an indication that using an EKF or a UKF to generate proposal distribution is effective in improving the performance of a particle filter. An algorithm that involves EKPF has a computational load of about 70% of that for an algorithm that involves UPF. Compared to EKF, it is apparent that the use of EKPF or UPF results in a much larger computational load.

The estimation results are much worse with the other algorithms. The average RMSEs obtained with GPF and APF are close to 90% of that for EKF. IMM algorithms that use GPF have comparable computational complexity with the individual GPF, but perform badly in the sense that there are instances of divergence. APF requires about 50% more computational load than GPF. The remaining algorithms have lower computational load than GPF and APF but yield little improvement over EKF, with average RMSEs about 95% of that for EKF.

Test results obtained with N_s increased in particle filters show no significant positive effects on the performance of the algorithms, with the exception of IMM algorithms

Filter A	RMSE (to 2 decimal places)						
	2925	3025	3125	3225	3325	Average	ϕ_A/ϕ_{EKF}
EKF	3.79e-03	3.22e-03	3.68e-03	2.63e-03	1.56e-03	2.98e-03	1.00
UKF	3.67e-03	3.10e-03	3.55e-03	2.48e-03	1.35e-03	2.83e-03	0.95
EKPF	8.76e-04	8.14e-04	9.45e-04	7.41e-04	8.79e-04	8.51e-04	0.29
GPF	3.21e-03	2.74e-03	3.19e-03	2.27e-03	1.39e-03	2.56e-03	0.86
RPF	3.62e-03	2.95e-03	3.56e-03	2.48e-03	1.42e-03	2.80e-03	0.94
SPF	3.57e-03	3.03e-03	3.59e-03	2.69e-03	1.61e-03	2.90e-03	0.97
UPF	9.40e-04	8.54e-04	1.01e-03	7.62e-04	8.44e-04	8.82e-04	0.30
APF	3.35e-03	2.85e-03	3.28e-03	2.32e-03	1.36e-03	2.63e-03	0.88
IEK	3.60e-03	3.06e-03	3.54e-03	2.50e-03	1.40e-03	2.82e-03	0.95
IUK	3.67e-03	3.10e-03	3.55e-03	2.48e-03	1.35e-03	2.83e-03	0.95
IEE	8.24e-04	7.36e-04	9.44e-04	6.62e-04	7.59e-04	7.85e-04	0.26
IER	3.65e-03	3.09e-03	3.57e-03	2.52e-03	1.43e-03	2.85e-03	0.96
IES	3.63e-03	3.09e-03	3.56e-03	2.53e-03	1.42e-03	2.85e-03	0.96
IEU	8.40e-04	7.86e-04	9.21e-04	7.46e-04	1.65e-03	9.88e-04	0.33
IEA	3.62e-03	3.08e-03	3.56e-03	2.51e-03	1.42e-03	2.84e-03	0.95
IUE	8.25e-04	7.21e-04	9.35e-04	6.62e-04	8.77e-04	8.04e-04	0.27
IUR	3.66e-03	3.09e-03	3.54e-03	2.48e-03	1.34e-03	2.82e-03	0.95
IUS	3.66e-03	3.09e-03	3.54e-03	2.48e-03	1.34e-03	2.82e-03	0.95
IUU	8.45e-04	7.83e-04	9.19e-04	8.24e-04	1.71e-03	1.02e-03	0.34
IUA	3.66e-03	3.09e-03	3.54e-03	2.48e-03	1.34e-03	2.82e-03	0.95
IEUE	8.30e-04	7.28e-04	9.36e-04	6.65e-04	7.52e-04	7.83e-04	0.26
IEUR	3.63e-03	3.07e-03	3.53e-03	2.47e-03	1.35e-03	2.81e-03	0.94
IEUS	3.63e-03	3.07e-03	3.53e-03	2.48e-03	1.35e-03	2.81e-03	0.94
IEUU	8.48e-04	7.73e-04	9.21e-04	7.39e-04	1.48e-03	9.53e-04	0.32
IEUA	3.64e-03	3.07e-03	3.53e-03	2.48e-03	1.36e-03	2.81e-03	0.95

Table 3.18: Errors in estimation of call option prices.

that involve GPF. From the additional numerical tests carried out, it is observed that when $N_s = 300$, average RMSE obtained with each of IEG and IEUG is about 95% of that obtained with EKF, while IUG still encounters occurrences of divergence. Further, when $N_s = 500$, IEG, IUG and IEUG each yields an average RMSE of about 95% of that for EKF.

2. Statistical comparison of filtering algorithms

As in Section 3.4.1.3, comparison of the performance (in terms of the mean square error in the estimation of call and put option prices) of the filtering algorithms implemented is formulated as a hypothesis testing problem as follows [13, Sections 1.5 and 11.5].

Let A be an arbitrary filtering algorithm implemented in the preceding simulation tests. Let J_A denote the actual mean square error in option price estimation. The

Filter A	RMSE (to 2 decimal places)						
	2925	3025	3125	3225	3325	Average	ϕ_A/ϕ_{EKF}
EKF	2.37e-03	2.65e-03	3.21e-03	4.05e-03	3.37e-03	3.13e-03	1.00
UKF	2.21e-03	2.49e-03	3.04e-03	3.95e-03	3.30e-03	3.00e-03	0.96
EKPF	7.43e-04	8.07e-04	9.07e-04	8.44e-04	9.64e-04	8.53e-04	0.27
GPF	2.10e-03	2.32e-03	2.81e-03	3.54e-03	3.06e-03	2.77e-03	0.88
RPF	2.20e-03	2.50e-03	3.00e-03	3.96e-03	3.29e-03	2.99e-03	0.95
SPF	2.22e-03	2.59e-03	2.95e-03	3.87e-03	3.25e-03	2.97e-03	0.95
UPF	8.06e-04	8.26e-04	9.69e-04	9.07e-04	1.02e-03	9.06e-04	0.29
APF	2.10e-03	2.34e-03	2.86e-03	3.67e-03	3.12e-03	2.82e-03	0.90
IEK	2.25e-03	2.51e-03	3.06e-03	3.91e-03	3.28e-03	3.00e-03	0.96
IUK	2.21e-03	2.49e-03	3.04e-03	3.95e-03	3.30e-03	3.00e-03	0.96
IEE	6.51e-04	7.61e-04	8.81e-04	7.96e-04	1.04e-03	8.25e-04	0.26
IER	2.29e-03	2.53e-03	3.07e-03	3.95e-03	3.29e-03	3.03e-03	0.97
IES	2.28e-03	2.54e-03	3.08e-03	3.94e-03	3.29e-03	3.03e-03	0.97
IEU	7.18e-04	7.94e-04	8.97e-04	8.46e-04	1.74e-03	9.98e-04	0.32
IEA	2.29e-03	2.55e-03	3.08e-03	3.95e-03	3.30e-03	3.03e-03	0.97
IUE	6.53e-04	7.60e-04	8.73e-04	8.06e-04	1.15e-03	8.48e-04	0.27
IUR	2.20e-03	2.48e-03	3.04e-03	3.95e-03	3.29e-03	2.99e-03	0.96
IUS	2.20e-03	2.48e-03	3.04e-03	3.95e-03	3.29e-03	2.99e-03	0.96
IUU	7.17e-04	8.06e-04	8.87e-04	8.98e-04	1.79e-03	1.02e-03	0.33
IUA	2.20e-03	2.48e-03	3.04e-03	3.95e-03	3.29e-03	2.99e-03	0.95
IEUE	6.57e-04	7.78e-04	8.76e-04	7.98e-04	9.88e-04	8.19e-04	0.26
IEUR	2.21e-03	2.47e-03	3.04e-03	3.93e-03	3.28e-03	2.99e-03	0.95
IEUS	2.21e-03	2.48e-03	3.03e-03	3.93e-03	3.28e-03	2.99e-03	0.95
IEUU	7.36e-04	7.87e-04	9.01e-04	8.59e-04	1.56e-03	9.68e-04	0.31
IEUA	2.21e-03	2.48e-03	3.04e-03	3.93e-03	3.28e-03	2.99e-03	0.95

Table 3.19: Errors in estimation of put option prices.

hypothesis testing problem is to test the null hypothesis

$$H_0 : \Delta = J_{EKF} - J_A \leq 0 \quad \text{(algorithm } A \text{ not better than EKF)}, \tag{3.69}$$

versus the alternate hypothesis

$$H_1 : \Delta = J_{EKF} - J_A > 0 \quad \text{(algorithm } A \text{ better than EKF)}, \tag{3.70}$$

subject to

$$\text{Prob}\{\text{accept } H_1 | H_0 \text{ is true}\} = \alpha \quad \text{(level of significance for hypothesis } H_0).$$

Let $(\hat{u}_A)_{ik}$ denote the estimate of the call (respectively, put) option price for algorithm A at the k-th scan in the i-th simulation run, $i = 1, \ldots, C$, $k = 1, \ldots, L$, and u_k be the actual call (respectively, put) option price at the k-th scan, $k = 1, \ldots, L$. With reference to Equations 3.48 and 3.58, the statistical test is based on the independent mean square error differences,

$$(\Delta_A)_i := \frac{1}{L}\sum_{k=1}^{L}\Big\{\big|(\hat{u}_{EKF})_{ik} - u_k\big|_2^2 - \big|(\hat{u}_A)_{ik} - u_k\big|_2^2\Big\}, \quad i = 1, \ldots, C, \tag{3.71}$$

which are independent from run to run. The test uses the sample mean of the above differences (Equation 3.49),

$$\bar{\Delta}_A := \frac{1}{C}\sum_{i=1}^{C}(\Delta_A)_i,$$

and its associated standard deviation (Equation 3.50),

$$\sigma_{\bar{\Delta}_A} := \Big\{\frac{1}{C^2}\sum_{i=1}^{C}\big((\Delta_A)_i - \bar{\Delta}_A\big)^2\Big\}^{\frac{1}{2}}.$$

As in Section 3.4.1.3, we use $\alpha = 0.05$ (equivalently, 5% level of significance). Thus, H_1 is accepted if the test statistic

$$\rho = \frac{\bar{\Delta}_A}{\sigma_{\bar{\Delta}_A}} > 1.65.$$

Tables 3.20 and 3.21 show the test for differences of mean square errors in state estimation between EKF and each of the other filtering algorithms.

It is observed that all the listed algorithms have statistically significant improvement over EKF (that is, test statistic $\rho > 1.65$) in the estimation of put option prices. For call options with strike price 3325, IEU, IUU and IEUU do not have improvement that is sufficient to reject the null hypothesis H_0 (at 5% level of significance). In addition, SPF does not provide sufficient justification to reject the null hypothesis H_0 (at 5% level of significance) for call options with strike prices 3225 and 3325.

The process model is linear Gaussian and the measurement model is not highly nonlinear in this problem. The IMM algorithms that use EKPF and UPF have insignificant or no superiority over the individual EKPF and UPF. In general, it is likely to have more complicated conditions such as high non-Gaussianity and/or nonlinearity in system models, as well as possibly large system noise. The IMM algorithms are more likely to perform better than the individual particle filters for such problems.

3.6 Filter Performance for Manœuvring Target Tracking and Modelling Financial Option Prices

In the target tracking problems discussed in Sections 3.4.1 and 3.4.2, the test trajectories considered are highly manœuvring, with variable behaviour types. Hence, the IMM

Filter A	Test statistic for call option price estimation				
	2925	3025	3125	3225	3325
UKF	12.86	10.57	17.33	12.39	14.79
EKPF	195.62	133.80	217.88	102.66	8.23
GPF	52.95	38.84	55.11	26.25	12.25
RPF	16.15	20.64	14.81	11.52	8.02
SPF	21.76	14.67	10.39	-4.40	-3.40
UPF	190.05	134.62	218.21	100.48	7.46
APF	43.57	30.03	47.74	24.73	13.76
IEK	17.57	11.39	14.86	9.58	10.25
IUK	12.86	10.57	17.33	12.39	14.79
IEE	193.46	132.72	208.67	104.25	25.14
IER	13.00	10.15	10.74	8.80	7.61
IES	14.24	10.32	13.13	7.57	8.49
IEU	192.47	136.10	222.73	102.54	-0.91
IEA	12.96	10.07	13.91	7.60	8.32
IUE	194.62	136.89	223.85	105.04	10.17
IUR	13.35	11.32	17.78	13.00	15.52
IUS	13.22	11.37	17.66	12.85	15.60
IUU	194.04	136.02	213.80	40.16	-1.17
IUA	13.50	11.35	17.90	12.98	15.52
IEUE	197.06	134.30	207.59	103.67	29.12
IEUR	16.58	12.63	18.37	12.65	13.70
IEUS	15.25	12.76	17.42	12.87	14.09
IEUU	193.46	132.28	216.34	101.65	0.91
IEUA	14.98	12.54	18.30	13.16	14.18

Table 3.20: Comparison of EKF with other filters in call option price estimation.

algorithm is suitable for modelling these problems. It appears that compared to IMM algorithm variants that use EKPF, UPF or GPF, those which use RPF, SPF or APF tend to perform better for manœuvring target tracking. This could be due to the prominent nonlinearity and/or non-Gaussianity in these problems, so a nonlinear filter (namely, EKF, UKF, EKPF, UPF or GPF) that obtains a Gaussian approximation to the proposal distribution or the posterior pdf of the system state may not be able to work well for these problems. It is also possible that poor state estimation or divergence occurs in each of the posed problems because of the accumulation of rounding errors when many mathematical operations need to be carried out. This is particularly evident when IMM algorithm variants use EKPF or UPF.

For the problem on modelling the prices of financial options, the process model is linear Gaussian and the measurement model is not highly nonlinear. The variation in the behaviour of the option prices is also moderate throughout the trading days. Hence,

Filter A	Test statistic for put option price estimation				
	2925	3025	3125	3225	3325
UKF	13.96	17.20	16.82	11.78	8.58
EKPF	93.99	128.36	155.06	229.00	57.15
GPF	21.24	33.57	39.14	48.62	35.45
RPF	13.85	13.70	18.46	9.47	8.43
SPF	11.88	6.70	23.78	20.79	11.70
UPF	94.25	127.45	151.72	222.87	48.61
APF	22.14	30.62	34.16	40.13	27.27
IEK	9.38	12.18	13.47	12.91	8.99
IUK	13.96	17.20	16.82	11.78	8.58
IEE	97.54	128.94	148.49	218.99	96.05
IER	6.10	10.97	11.42	9.75	8.22
IES	7.03	10.73	11.38	11.01	7.30
IEU	95.14	123.03	149.76	226.45	25.86
IEA	5.50	9.66	11.12	9.66	7.41
IUE	96.46	134.19	154.18	232.10	41.28
IUR	14.38	18.31	17.32	12.31	9.25
IUS	14.54	18.33	17.31	12.12	9.25
IUU	94.28	126.20	150.41	146.21	20.85
IUA	14.99	18.35	17.52	12.32	9.33
IEUE	96.48	126.31	151.99	223.22	135.25
IEUR	13.40	18.51	16.66	12.70	9.93
IEUS	13.79	17.06	16.87	13.05	9.74
IEUU	90.59	125.80	147.61	225.08	30.64
IEUA	12.97	17.89	16.73	13.33	10.27

Table 3.21: Comparison of EKF with other filters in put option price estimation.

the IMM algorithm variants generally do not have significant superior performance over the individual nonlinear filters for this problem. In particular, it is noted that IMM algorithm variants which use GPF have instances of divergence, while GPF does not. This could be due to the accumulation of rounding errors when effective state estimation cannot be achieved with EKF and/or UKF. Rectification can be done via an increase in the number of particles N_s used in GPF. It is apparent from the numerical results that in this problem, it is probably advantageous to use EKPF (respectively, UPF), which provides an indication that using an EKF (respectively, a UKF) to generate the proposal distribution is effective in improving the performance of a particle filter.

3.7 Summary

In this chapter, we have proposed a three-model IMM based algorithm for target tracking. Various combinations of extended Kalman filters, unscented Kalman filters and particle filters are used for the models. The proposed IMM variants are applied to two problems. The first problem is on 3D manœuvring target tracking. The second problem is on localization and tracking in a horizontal plane with a time difference of arrival system. Each IMM algorithm variant uses a constant velocity model, a constant acceleration model and a coordinated turn model.

In the first problem, IEK (IMM algorithm using EKF in every model) is used as the basis case for comparison of simulation results. Among the IMM algorithm variants implemented, IUR and IUS achieve better overall performance in state estimation (that is, smaller estimation errors). In the second problem, it is noted that IEK diverges. EKF is used as the basis case for comparing simulation results. Based on the analysis of numerical results, IUK, IUR, IUS, IUA, IEUR, IEUS and IEUA have better overall performance than EKF, in terms of smaller state estimation errors obtained.

Generally, a method chosen for problem solving is a trade-off between computational complexity and accuracy in results. Taking into consideration these two factors, it appears that IUR and IUS would be preferred over the remaining IMM algorithm variants. But they are not entirely consistent because in both test problems, it is observed that not all the state estimation errors are commensurate with the filter-calculated state error covariances. It has been mentioned that this is not an unusual phenomenon for adaptive algorithms [12], with timely adaptation being driven by the inconsistency. In practice, it is necessary to tune the filters to improve their respective consistency [13].

The proposed IMM algorithm variants, as well as extended Kalman filters, unscented Kalman filters and particle filters, are implemented for a problem on pricing financial options, in which real data is used for numerical tests. The performance of the filtering algorithms are analyzed and compared. It is observed that in terms of the state estimation errors obtained, EKPF and UPF have superior performance among the implemented filtering algorithms.

Finally, we give a brief discussion on the performance of the IMM algorithm variants and nonlinear filters for the problems on manœuvring target tracking and modelling the prices of financial options.

Chapter 4

Intent Inference for Air Defence and Conformance Monitoring

4.1 Introduction

The human brain has remarkable capabilities in perception and reasoning. However, the amount of complex data/information that can be processed by the human brain is constrained by the limited memory capacity. Hence, computational tools are necessary to provide cognitive aid to the human brain in attaining better performance in intellectual tasks, such as decision making.

Intent inference is about analyzing the actions and activities of an opponent or a target of interest to obtain a conclusion (prediction) on its purpose [16, 104, 177, 230]. Generally, data (collectively called observables) concerning the opponent are first collected from available sources. Next, the data are fused to obtain useful information. Finally, the fused information is utilized to derive the inferred intent of the opponent. It is desirable that intent inference be able to provide three kinds of hypotheses about an opponent's objective [16, 177]:

- Descriptive intent inference - provides insight into the motivations behind preceding actions;
- Predictive intent inference - anticipates the opponent's future actions given his deduced goals;
- Diagnostic intent inference - detects differences between predicted and observed actions to reveal possible errors.

Accurate prediction of an opponent's intention, actions and reactions would be useful for the purpose of devising effective responses to his actions, as well as planning for one's own operations.

Intent inference has been used in applications such as intelligent transportation systems (infer and detect a driver's intent [288]) and air traffic management (ATM) (predict the future trajectory of an air vehicle and the states of nearby aircraft [182,344]). Other applications include the medical domain, recommender systems, tutoring systems and team intent identification [177].

In this chapter, we report our research on two applications of intent inference [102, 234]. The first task is to determine the intent of the pilot (equivalently, the flight mission) of an aircraft being tracked by a military surveillance system [234]. The second involves conformance monitoring in air traffic control (ATC) systems [269].

Next, we compare the performance of IMM algorithm variants from Chapter 3 for state estimation within our proposed approach for intent inference. Finally, some issues concerning approach by multiple aircraft are briefly discussed.

This chapter is organized as follows. Section 4.2 provides a general discussion on intent inference and a brief review on related work from the research literature. Section 4.3 describes our proposed fuzzy logic approach for intent inference based on the analysis of flight profiles for attack aircraft. In addition, the environmental context of the tracked aircraft is taken into consideration during the execution of the inference process. The impact of this additional factor on the inference outcome is investigated. Four different test scenarios are used to evaluate the feasibility of the proposed method. Section 4.4 is focussed on conformance monitoring in ATC/ATM systems. Section 4.5 presents simulation tests and results. Section 4.6 compares the performance of IMM algorithm variants from Chapter 3 for the state estimation component in our proposed intent inference method. Section 4.7 gives a discussion on handling an approach by multiple aircraft. Section 4.8 provides a summary on this chapter.

4.2 Intent Inference

The *Boyd Control Loop* (also called *Boyd's Decision Loop* or the *Observe, Orient, Decide, and Act* (OODA) *Loop*) [123, 257] is a popular model that has been used for formalizing concepts of tactical command and decision making. It describes human and organizational behaviour as a continuous, iterative and cyclic process of *Observation* (represents event perception), *Orientation* (corresponds to the process of memory and cognition, the activity that provides environmental context and individual expectations), *Decision* (describes the process of cognitive comparison) and *Action* (equals the resulting behaviour). In particular, the function Orientation shapes the way the other functions, Observation, Decision and Action, are done.

The emphasis of this model is placed on shortening the cycle to perform the Observe to Act loop (see Figure 4.1):

- Observe - gather data from the environment via human and related senses,
- Orient - gain situation awareness and perform situation and impact/threat assessment based on the information derived from the data obtained,
- Decide - respond to situation and work out follow-up actions,
- Act - execute the planned response,

to the extent that the opponent cannot respond in time to carry out countermeasure, thus gaining superiority in the engagement. The OODA Loop can also be applied to computer-assisted cognition.

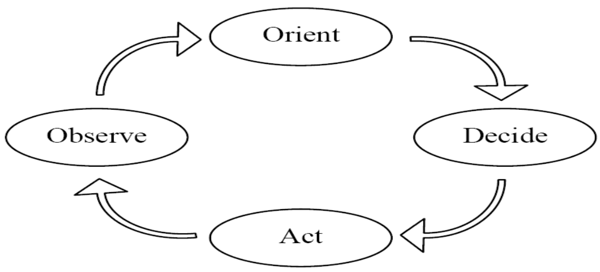

Figure 4.1: The OODA Loop.

An intent inference system provides reasoning about the opponent's intent, mission objective, or motivation. By nature of the inference mechanism, the intent inference system will also be able to provide prediction on the opponent's possible future actions or activity according to the inferred intent. Thus, it serves as useful decision support to the decision maker. In this way, the inference system not only contributes to better situation awareness and aids in resolving ambiguity that arises from multi-source fusion, but further unloads the decision maker and helps in shortening the decision making process.

Intent inference is a relatively young and challenging research area as compared to the maturing lower level data fusion. Emerging interest in the application of this research area can be found in the military arena [16, 290] and antiterrorism [144, 153]. Generally, intent and activity inference requires a cognitive architecture with knowledge-based modelling. Inputs to the inference system are information gathered through intelligent autonomous agents or provided by multiple sensing sources, including reports from human intelligence. Through modelling, the structure and pattern of opponent entities, as well as their behaviour and relationships, are captured. The focus of the inference mechanism is on contextual and relational reasoning as opposed to single entity reasoning at lower level fusion process. The inference mechanism may be based on a rule-based system or a more dynamic reasoning system such as Bayesian networks. In this thesis, a fuzzy inference system (FIS), also known as a fuzzy-rule-based system, is used.

4.2.1 Related Research Work

A method for pilot intent inference in real-time was investigated in [181]. It was based on plausible models of intent and a process for identifying models that matched observed aircraft motion best. The models were ranked based on their correlation with measured aircraft motion. The highest ranked plausible models of intent made up the best estimate of the aircraft intent. Sequences of actions were executed to infer guidance and navigation task intents of the tracked aircraft. The inferred intent was then used as a basis for trajectory prediction.

The authors of [339] proposed an intent-based trajectory prediction algorithm to carry out manœuvring aircraft tracking, aircraft intent inference and trajectory prediction. A hybrid estimation algorithm was used for estimating the states and flight mode of the aircraft. Intent inference was posed as a maximum likelihood problem. Pilot intent inference was obtained via the combination of the state and flight mode estimates, ATC regulations, the flight plan of the aircraft and environment information. The inferred intent and the aircraft motion (state and flight mode estimates) were used for the computation of trajectory prediction. The proposed algorithm was tested and analyzed through simulations in different scenarios representative of aircraft operations.

In [5], a hybrid system model of intent inference was constructed for air traffic controllers. An algorithm based on the Interacting Multiple Model (IMM) Kalman filter (the State Dependent Transition Hybrid Estimation algorithm) was implemented for state estimation, as well as the generation of residuals (discrepancies) between the observed aircraft states and the expected aircraft states. The residual mean was generated based on probabilistic methods. The proposed model was applied to an example problem on conformance monitoring. A statistical test was carried out on the residual means for both the conformance monitoring model and the actual aircraft system to obtain a conclusion/decision on conformance or non-conformance.

Conformance monitoring in ATC is a relatively new application of intent inference. Some research work based on fault detection has been done in this area [268–273] and will be discussed in Section 4.4.

4.2.2 Inference Mechanism

Classification is the process of inferring the concept behind an available collection of observations. This task covers any context in which some decision or forecast is made based on available information. It involves the establishment of a mapping from a measurement (an observation) space to a decision space. Input measurement/observation data is assigned into one or more predetermined classes based on the selection/extraction, as well as the processing or analysis, of significant features or attributes. Some commonly used approaches to classification are briefly discussed below [163].

4.2.2.1 Statistical Approach

Statistical (or decision theoretic) classifiers are generally characterized as having an explicit underlying probabilistic model. In a parametric classification procedure, a set of characteristic measurements (features) are extracted from the input data, and are used to assign each feature vector to one of the predetermined classes. Features are assumed to be generated by a state of nature, the underlying model represents a state of nature, set of probabilities, or probability density functions, that are conditional on the classes.

There are cases when there is insufficient prior information available, or when it is not necessary, to make assumptions about the distribution associated with the feature vector in the different classes. Under such circumstances, it is possible to use non-parametric estimation of the pdf involved to build distribution-free methods of classification (that is, non-parametric classifiers).

Statistical classifiers generally work reasonably well for problems in which structures are not deemed significant.

4.2.2.2 Neural Network Approach

A neural network assumes that a set of input data and their correct classifications are given. The architecture of a neural net includes layers of interconnected nodes. It is characterized by a set of weights and activation functions which determine the transmission of information from the input layer to the output layer. The training data is used to train the neural network and adjust the weights until the correct classifications are obtained. The complete network generally represents a complex set of interdependencies, which may incorporate an arbitrary degree of nonlinearity.

Neural networks are suitable for dealing with problems large amount of features and classes. They can be applied to problems that involve generalization, parallel processing, or discrimination among classes with highly nonlinear boundaries.

4.2.2.3 Fuzzy Logic Approach

Classification is often done with some degree of uncertainty. In problems with data that are noisy and distorted, complications can arise and lead to ambiguous situations in which classified data may belong in some degree to more than one class, or the classification outcome itself may be in doubt. Fuzzy logic (or fuzzy set theory) can be introduced to deal with such problems. In fuzzy classification, an input data entity is assigned a membership value in the interval $[0, 1]$ in each predetermined class.

4.2.3 Proposed Approach

We propose that intent inference be carried out via a fuzzy logic approach (conceptual information on fuzzy logic used in this thesis [152, 315, 316] is given in Section A.2). The main reasons that motivate the use of the proposed approach are as follows.

Firstly, compared to statistical and probabilistic methods used in most related research work, fuzzy logic techniques are particularly suitable for modelling problems with inherent imprecision properties [123, 229]. The problems to be discussed in this chapter involve observation/information associated with human cognitive processes such as thinking and reasoning, in which uncertainties and imprecision are usually inherent. Therefore, it is appropriate to use fuzzy logic to deal with these problems.

Secondly, fuzzy logic techniques are useful for the fusion of information from multiple input sources and the application of heuristics to determine the overall status of the inputs [64]. Hence, for each problem in this chapter, the information obtained from tracking the subject aircraft can be fused to determine the pilot intent, which is required by the surveillance/monitoring system users concerned for decision making.

Thirdly, implementation of fuzzy logic is simple, fast and efficient [208, 315]. This would be useful for problems in which computational load/time is a critical factor, such as the two problems of interest here. For the first task on air defence, it is essential to take pre-emptive action against potential adversaries as quickly as possible, in order to avert possible attacks. For the second problem on conformance monitoring in ATC systems, it is important to minimize the delay in correcting any deviant aircraft behaviour that is detected.

A fuzzy inference system is a computing framework based on the concepts of fuzzy set theory, fuzzy rules and fuzzy reasoning (an inference procedure which derives conclusions from a set of fuzzy rules and available information) [152]. The basic structure of a fuzzy inference system comprises three conceptual components:

- rule base - contains a selection of fuzzy rules,
- database - defines the membership functions used in the fuzzy rules,
- reasoning mechanism - performs the inference procedure upon the rules and known facts to derive a reasonable ouptut or conclusion.

The inference mechanism used in this thesis is based on the widely accepted Mamdani's fuzzy inference method [152], which was one of the first control systems built using fuzzy set theory. It was proposed as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators.

The Mamdani-type FIS used here is generated using the MATLAB Fuzzy Logic Toolbox [315, 316]. The fuzzy inference process has five parts, namely, fuzzification of the input variables, application of the fuzzy operator in the antecedent, implication from the antecedent to the consequent, aggregation of the consequents across the rules, and defuzzification. Details on each part of the fuzzy inference process implemented for the two applications discussed in this chapter are provided in Sections 4.3 and 4.4.

4.3 Weapon Delivery by Attack Aircraft

Effective intent inference will greatly enhance the defence capability of a military force in taking pre-emptive action against potential adversaries. It serves as a form of advance warning in the prevention of a crisis (for instance, enemy attack) or facilitates the moderation of the impact of such a crisis. For a air defence system, the ability to accurately infer the likelihood of a weapon delivery by an attack aircraft is critical.

The type of weapon delivery for attack aircraft considered in this thesis is offset pop-up delivery. The definitions for some terms pertaining to this form of weapon delivery are stated below. Section 4.3.1 provides a brief description of offset pop-up delivery [319].

- Pop Point (PUP) - a position at which the pop-up attack is initiated, the point where climb is initiated.
- Pull-Down Point (PDP) - a manœuvre point where one transitions from the climbing to the diving portion of a pop-up delivery.
- Apex - the highest altitude in the pop-up delivery profile.
- Track Point (TP) - the starting point of tracking prior to arriving at planned release altitude.
- Release Point (RP) - the point at which weapon is released.

A tracked aircraft is considered to have constant speed, with the velocity components in the horizontal plane (parallel to ground) and the vertical axis (parallel to altitude) varying in different phases of the trajectory. In this application, altitude, distance and velocity are measured in feet above ground level (AGL), feet and knots respectively, unless otherwise stated.

4.3.1 Typical Offset Pop-up

The pop-up approach heading, as shown in Figure 4.2 [67], is at an angle (varies with the planned climb angle) from 15° to 90° from the final attack heading. This allows the pilot to acquire the target as soon as possible and maintain visual contact until weapon delivery is completed.

The pilot initiates the pop-up over a preplanned pop point at a minimum airspeed of 450 knots calibrated airspeed (KCAS). He selects his desired power, makes a 3-4 G wings-level pull to the desired climb angle and initiates a chaff/flare program. After popping, he has to maintain the planned climb angle and monitor the altitude gained.

When approaching the preplanned pull-down altitude, the pilot makes an unloaded roll in the direction of the target. He then performs a 3-5 G pull-down to intercept the planned dive angle. Interception of the planned dive angle while pointed at the aim-off point is a critical factor in attaining preplanned delivery parameters. It is usually acceptable to have minor deviations in the attack heading.

During the manœuvre, corrections are made to compensate for minor errors in the pop point or unexpected winds in the climb to the apex at the planned altitude. The planned apex altitude is normally achieved about half way through the pull-down manœuvre.

For safety reasons, a pilot would most probably abort a pop-up attack immediately if at least one of the following conditions arises:

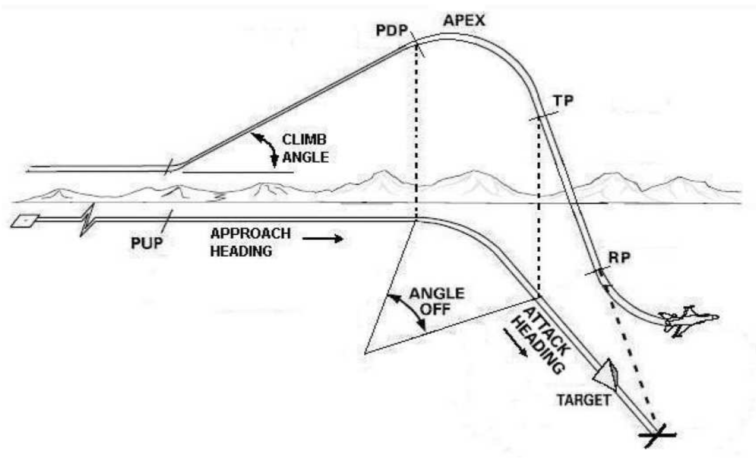

Figure 4.2: Flight profile for offset pop-up delivery.

- the actual dive angle exceeds the planned one by more than 5°,
- the airspeed goes below 350 KCAS (300 KCAS above 10000 feet AGL).

The occurrence of such conditions would result in inaccuracy in the impact point of the released weapon.

4.3.2 Process and Techniques

Our proposed procedure for inferring the possibility of weapon delivery by a tracked attack aircraft, based on flight profiles, is given below.

Procedure 1

1. For an aircraft being tracked, record its state information (sensor measurement data) through observation.
2. Apply the IMM algorithm [222, 230] to update the track state estimates.
3. For each track state estimate, use the position components to identify the environmental context and hence the corresponding location sensitivity index (LSI) (details in Sections 4.3.2.1 and 4.5.1).
4. Fuzzy inference process
   a. Input
      i. relevant parameters of the filtered flight trajectory, and
      ii. LSI obtained in Step 3,

   to a Mamdani-type fuzzy inference system generated using the MATLAB Fuzzy Logic Toolbox [315, 316].
   b. Output produced by the FIS is the inferred possibility of weapon delivery by the tracked aircraft.

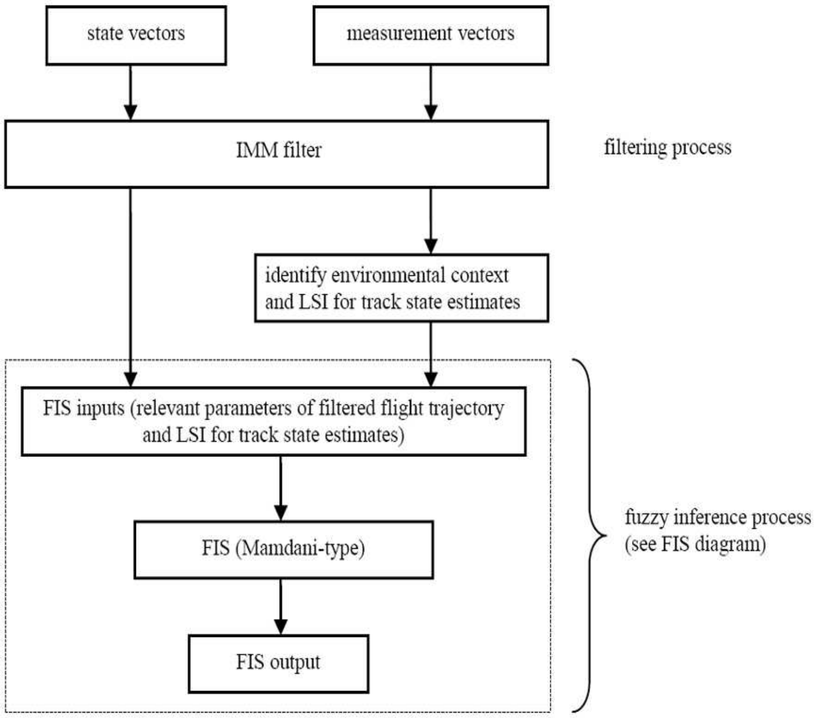

Figure 4.3: Overview of proposed system.

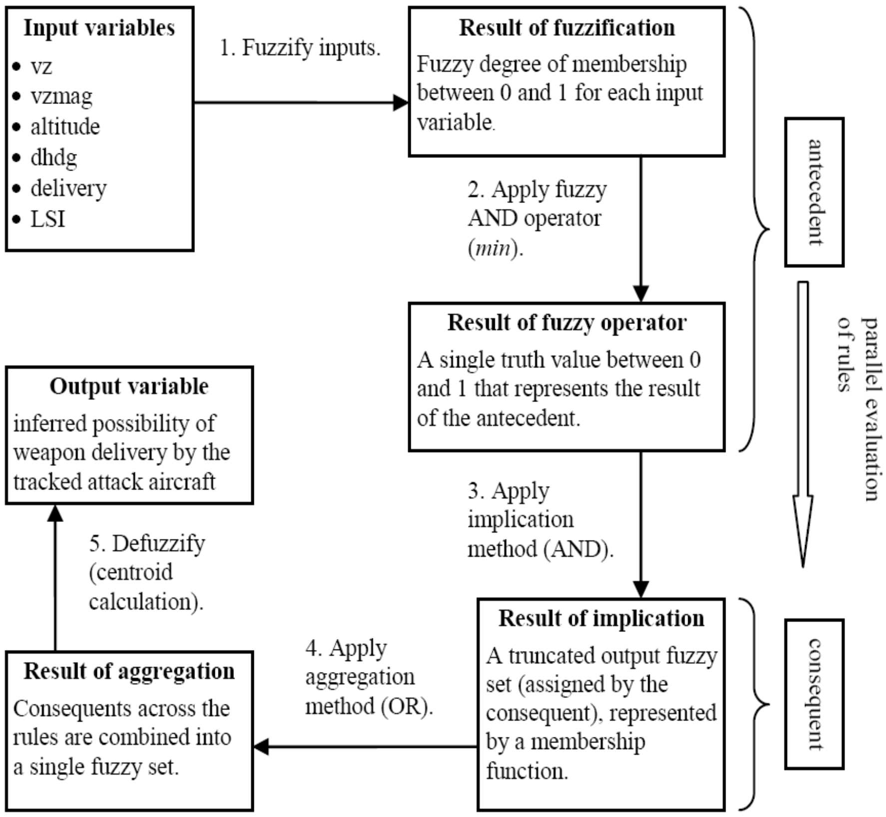

Figure 4.4: Fuzzy inference system.

An overview of the system for the proposed approach is shown in Figure 4.3. The entire fuzzy inference process is shown in Figure 4.4. The following subsections provide details on the fuzzy inference process.

4.3.2.1 Fuzzification of the Input Variables

In the first step, each input variable is a non-fuzzy/crisp numerical value within its universe of discourse and is assigned a linguistic value in the interval $[0, 1]$ via a membership function. The input variables considered in the current application are obtained from kinematic parameters of the filtered flight trajectory. Elaboration on each of the input variables, with respect to the tracked aircraft, is given below.

The first variable is the velocity along the vertical axis (abbreviated *vz*). It is classified as either positive (denoted by "> 0") or negative (denoted by "< 0"), indicating either upward or downward motion respectively. The second variable is the magnitude of vz (abbreviated *vzmag*). The third variable is the altitude. The fourth variable is an indicator for the occurrence of a change in heading (measured in radians, abbreviated *dhdg*) during the time interval between consecutive scans. A change in heading is considered to have occurred when the difference in heading between two consecutive records along the filtered flight trajectory exceeds a chosen threshold value ($\pi/180$ radians in the current application). The fifth variable is an indicator for the likelihood of a weapon delivery (abbreviated *delivery*) by the tracked aircraft. A weapon delivery is considered unlikely when at least one of the following conditions occurs:

- the actual dive angle exceeds the planned one by more than $5°$,
- the airspeed goes below 350 KCAS (300 KCAS above 10000 feet AGL).

The sixth variable is an index representation of the environmental context of the tracked aircraft, named *location sensitivity index* (abbreviated *LSI*). The LSI is based on the degree of sensitivity of the spatial domain in which the tracked aircraft is travelling. High LSI corresponds to highly sensitive locations, including vicinities of critical infrastructure such as government establishments. Low LSI corresponds to locations with low sensitivity, including regions that are remote or not habitable.

Figures 4.5 to 4.10 show the membership functions for the six input variables. Table 4.1 shows the symbols and their corresponding linguistic values for membership functions (where applicable).

The number of levels for the linguistic values for membership functions can vary

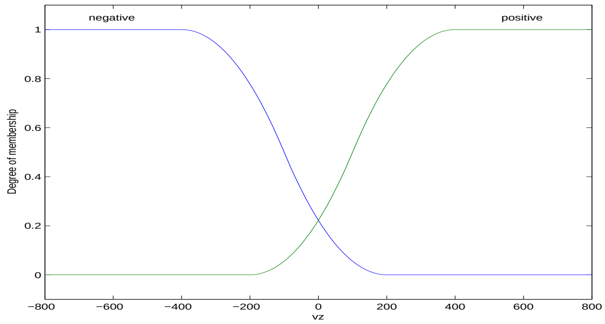

Figure 4.5: Membership functions of “vz”.

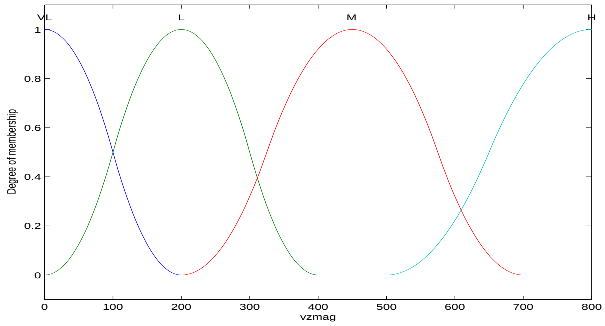

Figure 4.6: Membership functions of “vzmag”.

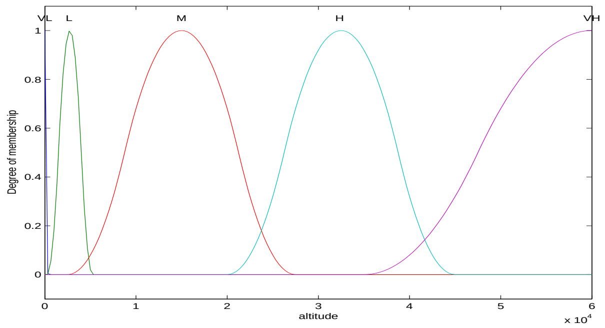

Figure 4.7: Membership functions of “altitude”.

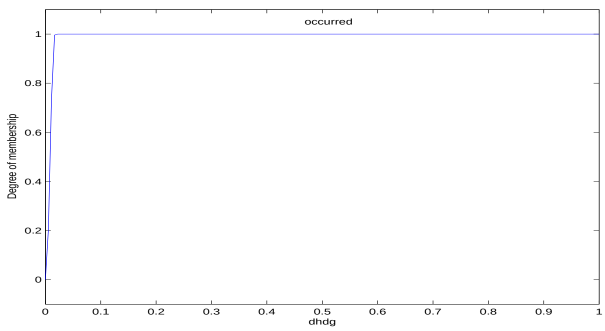

Figure 4.8: Membership function of “dhdg”.

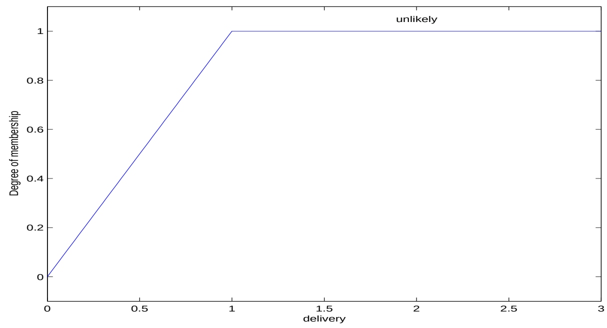

Figure 4.9: Membership function of "delivery".

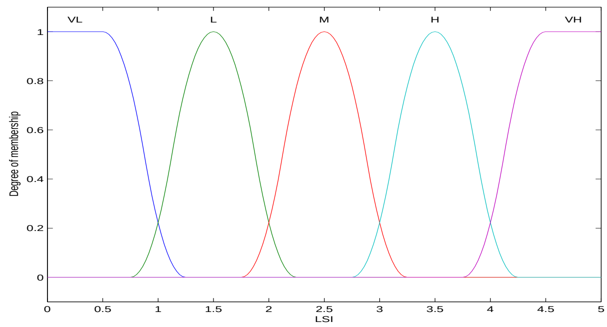

Figure 4.10: Membership functions of "LSI".

according to the amount of information available. Labels that are more descriptive can be used for various levels of linguistic values of a variable. An example is to use words such as fast, slow and constant when labelling different degrees of membership for variables related to velocity/speed.

4.3.2.2 Application of Fuzzy Operators

After fuzzification of the inputs, the degree to which each part of the antecedent is satisfied for each rule is known. When an antecedent of a given rule has multiple parts, a fuzzy operator (such as those defined in Section A.2) has to be applied to the multiple membership values from fuzzified input variables, in order to obtain one single truth value. This output value (which lies in $[0, 1]$) represents the result of that antecedent for that rule and will be applied to the output function.

Symbol	VL	L	M	H	VH
Linguistic value	Very Low	Low	Medium	High	Very High

Table 4.1: Symbols used for membership functions.

4.3.2.3 Application of Implication Method

For each rule, apply a weight (1 is used in this thesis) to the single truth value given by the antecedent. Then implement the implication on this weighted value using the built-in AND method: min (minimum) function [315, 316]. The implication process yields an output fuzzy set (assigned by the consequent) which is truncated to the level of the weighted truth value of the antecedent. The rules used in the current application are listed in Table 4.2.

R1.	(altitude is VL) → (pos is VL).
R2.	($vz > 0$) & (dhdg is NOT occurred) & (LSI is VL) → (pos is L).
R3.	($vz > 0$) & (dhdg is NOT occurred) & (LSI is L) → (pos is L).
R4.	($vz > 0$) & (dhdg is NOT occurred) & (LSI is M) → (pos is M).
R5.	($vz > 0$) & (dhdg is NOT occurred) & (LSI is H) → (pos is M).
R6.	($vz > 0$) & (dhdg is NOT occurred) & (LSI is VH) → (pos is H).
R7.	($vz > 0$) & (vzmag is L) & (dhdg is occurred) & (LSI is VL) → (pos is L).
R8.	($vz > 0$) & (vzmag is L) & (dhdg is occurred) & (LSI is L) → (pos is M).
R9.	($vz > 0$) & (vzmag is L) & (dhdg is occurred) & (LSI is M) → (pos is M).
R10.	($vz > 0$) & (vzmag is L) & (dhdg is occurred) & (LSI is H) → (pos is H).
R11.	($vz > 0$) & (vzmag is L) & (dhdg is occurred) & (LSI is VH) → (pos is H).
R12.	($vz > 0$) & (vzmag is VL) & (dhdg is occurred) & (LSI is VL) → (pos is M).
R13.	($vz > 0$) & (vzmag is VL) & (dhdg is occurred) & (LSI is L) → (pos is M).
R14.	($vz > 0$) & (vzmag is VL) & (dhdg is occurred) & (LSI is M) → (pos is H).
R15.	($vz > 0$) & (vzmag is VL) & (dhdg is occurred) & (LSI is H) → (pos is H).
R16.	($vz > 0$) & (vzmag is VL) & (dhdg is occurred) & (LSI is VH) → (pos is VH).
R17.	($vz < 0$) & (altitude is NOT VL) & (delivery is NOT unlikely) & (LSI is VL) → (pos is M).
R18.	($vz < 0$) & (altitude is NOT VL) & (delivery is NOT unlikely) & (LSI is L) → (pos is H).
R19.	($vz < 0$) & (altitude is NOT VL) & (delivery is NOT unlikely) & (LSI is M) → (pos is H).
R20.	($vz < 0$) & (altitude is NOT VL) & (delivery is NOT unlikely) & (LSI is H) → (pos is VH).
R21.	($vz < 0$) & (altitude is NOT VL) & (delivery is NOT unlikely) & (LSI is VH) → (pos is VH).
R22.	($vz < 0$) & (delivery is unlikely) & (LSI is VL) → (pos is L).
R23.	($vz < 0$) & (delivery is unlikely) & (LSI is L) → (pos is M).
R24.	($vz < 0$) & (delivery is unlikely) & (LSI is M) → (pos is M).
R25.	($vz < 0$) & (delivery is unlikely) & (LSI is H) → (pos is H).
R26.	($vz < 0$) & (delivery is unlikely) & (LSI is VH) → (pos is H).

Table 4.2: Rules for fuzzy inference system (weapon delivery by attack aircraft).

Figure 4.11 shows the membership functions for the output variable (inferred possibility of weapon delivery by the tracked attack aircraft, abbreviated *pos*). The complexity of the rules can be modified according to the amount of information available.

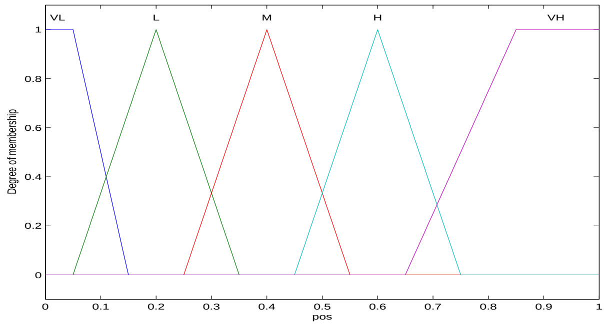

Figure 4.11: Membership functions of "pos".

4.3.2.4 Aggregation of All Outputs

It is necessary to determine an approach to combine the rules in an FIS in order to reach a decision/conclusion. The output fuzzy sets of each rule (obtained via the preceding implication method) are unified to form a single output fuzzy set, whose membership function assigns a weighting for every output value. The aggregation process inputs are the truncated output membership functions returned by the preceding implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable. This thesis utilizes the built-in OR method: max (maximum) function [315, 316] for the aggregation process. Therefore, the final membership function value is given by the maximum value among the consequent membership function values for each of the rules in the FIS.

4.3.2.5 Defuzzification

In the last step of the fuzzy inference process, let F denote the output fuzzy set of the preceding aggregation process and Z denote the universe of discourse that F is in. Let $\mu_F(\cdot)$ be the aggregated output membership function representing F. Defuzzification of F yields the output of the FIS, which is a single crisp/non-fuzzy number [152]. The built-in method of centroid calculation [315, 316] is used in this thesis. The defuzzified output, z_{COA}, is the center of area under $\mu_F(\cdot)$, defined by

$$z_{COA} = \frac{\int_Z \mu_F(z) z dz}{\int_Z \mu_F(z) dz}.$$

4.4 Conformance Monitoring

In conventional ATC and ATM operations, the controller creates a visualization of the current and future state dynamics of all aircraft under his control. For each individual aircraft, the controller determines if its observed behaviour conforms to the expected or planned path [268,273]. Unintentional deviations can result from noise in the surveillance systems, atmospheric effects and dynamics of the aircraft navigation systems. Such deviations can be used as threshold values in the definition of a "conformance region". An observed flight profile that lies within the region would be considered conforming, while one that lies beyond the region would be considered non-conforming. In the latter case, knowledge of the conformance status provides a basis for the air traffic controller to implement rectifying measures for the aircraft concerned.

In [269], an analysis framework was developed for the purpose of investigating issues pertaining to conformance monitoring in ATC/ATM. The conformance monitoring task was put forward as a fault detection problem. Fault detection and isolation techniques were used to determine if observable aircraft states were consistent with behaviour that was normal (that is, conforming) or abnormal (that is, non-conforming). In other words, non-conforming behaviour of an aircraft was regarded as a "fault" to be detected in the ATC/ATM system. The proposed framework comprised the following components:

- conformance basis -
  basis from which expected state behaviours of an aircraft are generated and against which observed behaviours of the subject aircraft are compared;

- actual system representation -
  key elements that execute instructions that form the communicated conformance basis;

- conformance monitoring model -
  generates expected state behaviours against which observed state behaviours are to be compared (requires appropriate level of fidelity to carry out effective conformance monitoring);

- conformance monitoring functions -
  determine at any time if observed state behaviours are consistent with expected state behaviours that are output by the conformance monitoring model.

The framework was implemented for several conformance monitoring tasks in ATC [270–272].

Enhancement and/or improvement of techniques for conformance monitoring is of much interest because of its importance in proper operation of ATC/ATM systems. In addition, there is much awareness of potential hazards to the air transport system posed by non-conforming aircraft that deviate from expected traffic patterns.

In order to maintain the safety, security and efficiency of ATC/ATM systems, timely detection of non-conforming behaviour in aircraft is essential. Our objective in this application is to use a fuzzy inference approach to determine if a tracked aircraft is navigating within conformance limits.

4.4.1 Process and Techniques

The proposed procedure (a slight modification of Procedure 1 in Section 4.3.2) for inferring the possibility of non-conformance in the behaviour of a tracked aircraft is stated below.

Procedure 2

1. For an aircraft under surveillance, record its state information (sensor measurement data) through observation.
2. Apply the IMM algorithm [222, 230] to update the track state estimates.
3. Fuzzy inference process
   a. Input relevant parameters of the filtered flight trajectory to a Mamdani-type fuzzy inference system generated using the MATLAB Fuzzy Logic Toolbox [315, 316].
   b. Output produced by the FIS is the inferred possibility of non-conformance in the behaviour of the tracked aircraft.

The system diagram for the proposed approach is identical to that shown in Figure 4.3, omitting the consideration of environmental context. Figure 4.4 shows the fuzzy inference process, with input and output variables replaced by those described in Section 4.4.1.1.

4.4.1.1 Fuzzy Inference Process

Firstly, fuzzification of the input variables is as described in Section 4.3.2.1. The input variables considered in the current application are obtained from kinematic parameters of the filtered flight trajectory. Each of the input variables, with respect to the tracked aircraft, is defined below.

The first variable is the deviation of the estimated position from the planned position (measured in feet, abbreviated dp). The second variable is the deviation of the estimated velocity from the planned velocity (measured in feet per second, abbreviated

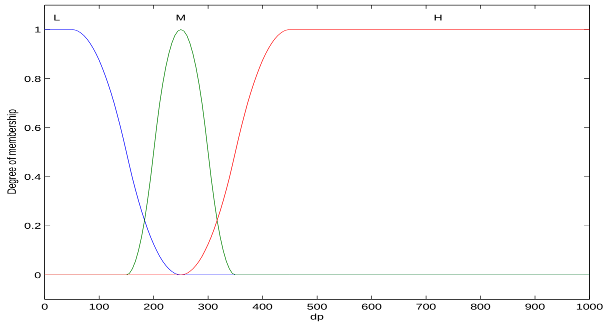

Figure 4.12: Membership functions of “dp”.

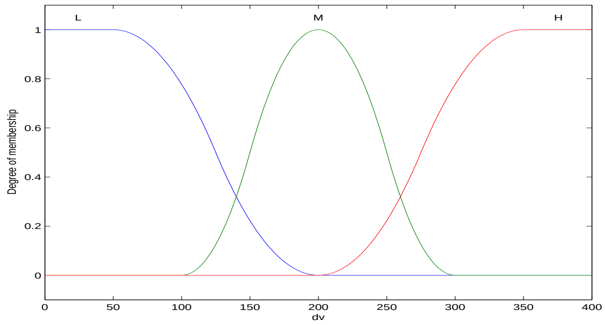

Figure 4.13: Membership functions of “dv”.

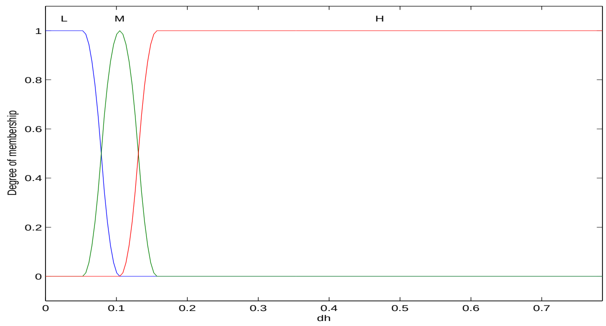

Figure 4.14: Membership functions of “dh”.

dv). The third variable is the deviation of the estimated heading from the planned heading (measured in radians, abbreviated dh).

Figures 4.12 to 4.14 show the membership functions for the three input variables. The symbols and their corresponding linguistic values for membership functions are shown in Table 4.1 (where applicable).

Next, rule evaluation (application of the fuzzy operator in the antecedent, followed by implication from the antecedent to the consequent) is carried out as stated in Sections 4.3.2.2 and 4.3.2.3. The rules used in the current application are listed in Table 4.3.

Figure 4.15 shows the membership functions for the output variable (inferred possibility of non-conformance in the behaviour of the tracked aircraft, abbreviated pnc).

R1.	(dp is L) & (dv is L) & (dh is L) → (pnc is L)
R2.	(dp is L) & (dv is L) & (dh is M) → (pnc is M)
R3.	(dp is L) & (dv is L) & (dh is H) → (pnc is M)
R4.	(dp is L) & (dv is M) & (dh is L) → (pnc is M)
R5.	(dp is L) & (dv is M) & (dh is M) → (pnc is M)
R6.	(dp is L) & (dv is M) & (dh is H) → (pnc is H)
R7.	(dp is L) & (dv is H) & (dh is L) → (pnc is M)
R8.	(dp is L) & (dv is H) & (dh is M) → (pnc is H)
R9.	(dp is L) & (dv is H) & (dh is H) → (pnc is VH)
R10.	(dp is M) & (dv is L) & (dh is L) → (pnc is M)
R11.	(dp is M) & (dv is L) & (dh is M) → (pnc is M)
R12.	(dp is M) & (dv is L) & (dh is H) → (pnc is H)
R13.	(dp is M) & (dv is M) & (dh is L) → (pnc is M)
R14.	(dp is M) & (dv is M) & (dh is M) → (pnc is H)
R15.	(dp is M) & (dv is M) & (dh is H) → (pnc is VH)
R16.	(dp is M) & (dv is H) & (dh is L) → (pnc is M)
R17.	(dp is M) & (dv is H) & (dh is M) → (pnc is H)
R18.	(dp is M) & (dv is H) & (dh is H) → (pnc is VH)
R19.	(dp is H) & (dv is L) & (dh is L) → (pnc is M)
R20.	(dp is H) & (dv is L) & (dh is M) → (pnc is H)
R21.	(dp is H) & (dv is L) & (dh is H) → (pnc is VH)
R22.	(dp is H) & (dv is M) & (dh is L) → (pnc is M)
R23.	(dp is H) & (dv is M) & (dh is M) → (pnc is H)
R24.	(dp is H) & (dv is M) & (dh is H) → (pnc is VH)
R25.	(dp is H) & (dv is H) & (dh is L) → (pnc is H)
R26.	(dp is H) & (dv is H) & (dh is M) → (pnc is VH)
R27.	(dp is H) & (dv is H) & (dh is H) → (pnc is VH)

Table 4.3: Rules for fuzzy inference system (conformance monitoring).

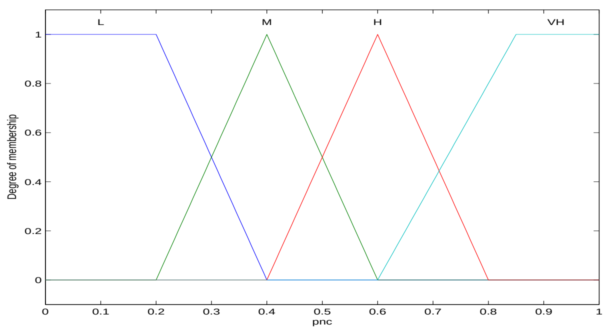

Figure 4.15: Membership functions of “pnc”.

As mentioned in Sections 4.3.2.4 and 4.3.2.5, the output fuzzy sets (assigned by the consequents) of each rule are unified to form a single output fuzzy set via an aggregation process. Defuzzification of this final output fuzzy set yields the output of the FIS, which is a single crisp/non-fuzzy number.

4.5 Simulation Tests and Results

We carry out simulation tests to verify the plausibility of the proposed approach. The state estimation component of the method is as follows. Consider a three-dimensional kinetic model described by the discrete-time dynamic system

$$X_{k+1} = f(X_k, w_k), \tag{4.1}$$

and the measurement/observation equation

$$Z_{k+1} = h(X_{k+1}, v_{k+1}). \tag{4.2}$$

At time step k, the state vector is $X_k = [x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k]^T$. The process noise vector w_k is assumed to be white Gaussian with covariance matrix Q. The measurement vector is Z_k and the measurement noise vector v_k is assumed to be white Gaussian with covariance matrix R. Scalar matrices are used for Q and R. The sampling interval is $T = 1$ (second).

The IMM algorithm used in this section corresponds to IEK from Table 3.1 (detatils in Section 3.3.4). It comprises a constant velocity model and two coordinated turn models (one left-turn and one right-turn). The transition probability matrix and the initial mode probability are

$$\begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0.9 & 0.05 & 0.05 \end{bmatrix}$$

respectively. The choices made for the transition probability matrix values [13, 265] are based on the following reasons. The frequency of mode switches for a tracked target is expected to be low, compared to that of it staying in the same mode (that is, remaining in the same type of motion). The probability of a switch from the current mode to another is expected to be the same for each of the remaining modes. The expected sojourn time of the system in the CV mode is likely to be higher than in the other modes. In addition, the two CT models only differ in their turning directions, so the transition probabilities for them are set in the same way.

4.5.1 Weapon Delivery by Attack Aircraft

We use the simulation results for the following test examples to evaluate the effectiveness of the proposed method.

Example 1: Aircraft in surveillance region of low to high LSI.

We use computation formulas in [319] to determine pop-up delivery parameters. Simulation is carried out on 100 different flight trajectories which are generated using various pop-up delivery parameter values.

For each test, as described in Procedure 1 (see Section 4.3.2), the IMM algorithm is applied to update the state vectors obtained from each flight trajectory. In the filter used, the discrete-time dynamic system of each model is of the form represented by Equations 4.1 and 4.2. Next, for each state estimate, determine the environmental context and the corresponding location sensitivity index.

Let A denote the xy-plane (horizontal plane) portion of the entire surveillance region, with the navigation convention (azimuth $= 0$ along the positive y-axis). Consider the partition

$$A = \bigcup_{i=1}^{2} \bigcup_{j=1}^{8} A_{ij},$$

where

A_{1j} is the j-th octant with $x^2 + y^2 < B_2^2$, $j = 1, \ldots, 8$,

A_{2j} is the j-th octant with $B_2^2 \leq x^2 + y^2 < B_1^2$, $j = 1, \ldots, 8$, and

bounds B_1 and B_2 are given positive constants.

The environmental contexts of the partition subsets of A are predetermined and can vary. Let M be a given matrix corresponding to the partition of A, where the LSI for each partition subset A_{ij} is $M(i, j)$, $i = 1, 2$, $j = 1, \ldots, 8$. Figure 4.16 shows the layout for A, with each partition subset denoted according to its subscript by (i, j), $i = 1, 2$, $j = 1, \ldots, 8$. For each state estimate X_k of the flight trajectory obtained from the filtering process, use the position components x_k and y_k to identify the partition subset, $A_{i(k),j(k)}$, that X_k is in and the corresponding LSI, $M(i(k), j(k))$. The relevant parameters of the flight trajectory obtained from the filtering process and the LSI obtained for the track state estimates are input to a Mamdani-type fuzzy inference system generated using the MATLAB Fuzzy Logic Toolbox [315, 316]. The output produced by the FIS is the inferred possibility of the tracked aircraft carrying out a weapon delivery. In this application, we propose to classify a tracked aircraft as having adversarial intent when the FIS output exceeds 0.85.

Figure 4.17 shows typical results obtained at different phases of the filtered flight trajectory (lower graph), in a scenario where the tracked aircraft travels from regions of low to high sensitivity (and LSI). In the upper graph, the solid curve shows the FIS output values (denoted by P henceforth, in this and subsequent test examples) obtained

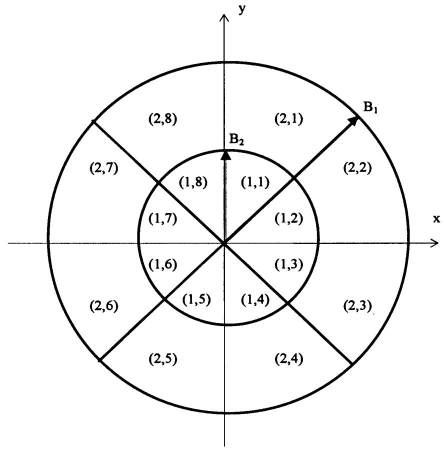

Figure 4.16: Partition of surveillance region (xy-plane).

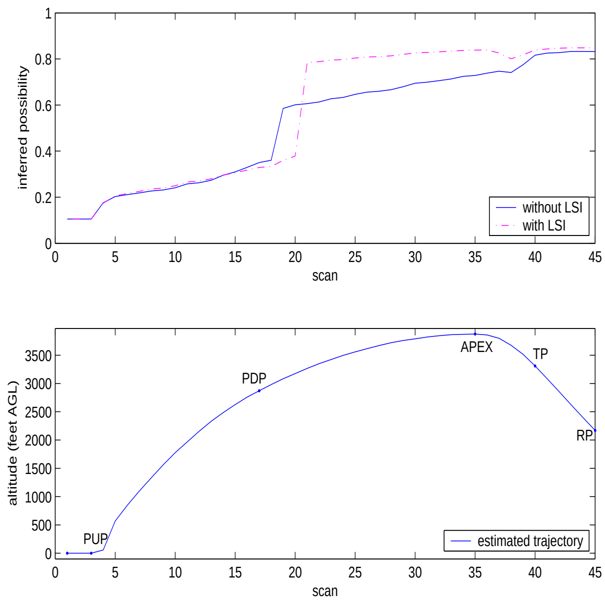

Figure 4.17: Example 1 - Fuzzy inference system output.

with only the flight profile considered during simulation. The dash-dot curve shows the FIS output values (denoted by P' henceforth, in this and subsequent test examples) obtained via simulation with both the flight profile and the environmental context of the tracked aircraft considered. Table 4.4 shows P and P' corresponding to the five specific points (defined in Section 4.3) on the filtered flight trajectory.

It can be observed from Figure 4.17 that P increases as time passes during the tracking process. The surge in P at scan 19 is triggered by motion that is characterized/interpreted by the FIS as the onset of transition from the climbing to the diving portion of a pop-up delivery. Thus, the FIS returns a significant increase in P, for warning purposes. P attains its peak around (and beyond) the apex. P remains high in the

Position on flight profile	PUP	PDP	Apex	TP	RP
Corresponding LSI for position	1	1	5	5	5
Without LSI	0.105	0.350	0.728	0.816	0.832
With LSI	0.105	0.329	0.838	0.839	0.848

Table 4.4: Example 1 - Fuzzy inference system output (to 3 decimal places).

later part of the tracking process. This observation provides verification for the feasibility of our proposed approach for adversarial intent inference, based on the assumption that the aircraft is approaching its weapon release point.

In the current scenario, the aircraft travels in regions of low sensitivity during the first 20 scans of the tracking process. Subsequently, the aircraft travels in regions of high sensitivity. In regions of low (respectively, high) sensitivity, low (respectively, high) corresponding LSI brings about $P' < P$ (respectively, $P' > P$). In the latter situation, the higher P' is likely to be useful in raising military defenders' alert against a potential adversary.

It appears from the simulation results that a tracked aircraft is very likely to carry out a weapon delivery when P (or P') exceeds 0.85. It is probably appropriate for military defenders to raise the level of vigilance when P (or P') exceeds 0.7. This would allow them to have more time to devise and take pre-emptive action against the potential adversary. Figure 4.17 shows that P' exceeds 0.7 earlier than P. This provides justification that taking into consideration the environmental context of the tracked aircraft is useful for improving the efficiency of our approach for adversarial intent inference.

Example 2: Aircraft in surveillance region of low LSI.

This example is analogous to Example 1, with the entire surveillance region being of low LSI. Typical simulation results obtained are shown in Figure 4.18.

The shapes of the plotted curves are similar to the corresponding ones in Figure 4.17. During the early stages of tracking, P and P' are low and almost identical. As in Example 1, there is a surge in P at scan 21, which is triggered by motion that is interpreted by the FIS as the onset of transition from climbing to diving portion of a pop-up delivery. Towards the later part of the tracking process, P exceeds 0.7, which is reasonably high. On the other hand, $P' < P$ and remains below 0.6, which is moderate. In addition, P does not exceed 0.75, which is below the proposed threshold value of 0.85 for classifying an aircraft as having adversarial intent.

Compared to Example 1, there appears to be less critical need/urgency in taking

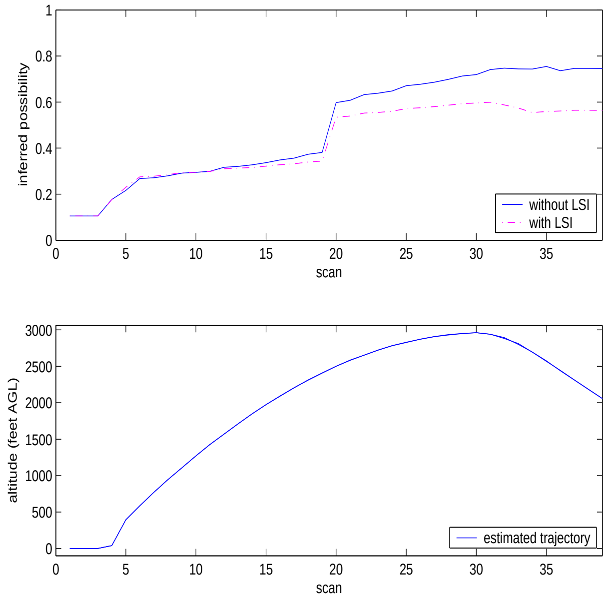

Figure 4.18: Example 2 - Fuzzy inference system output.

action against the tracked aircraft. This is due to the low sensitivity in the surveillance region, which leads to relatively lower P' values when corresponding P values become high. However, it would probably be advisable for the defenders to maintain their vigilance against such an aircraft, whose flight profile closely resembles that of a pop-up delivery.

Example 3: Aircraft cruising at high altitude.

We consider an aircraft that cruises at high altitude throughout the approach. Two possible scenarios are described as follows.

Example 3a: Aircraft cruising in surveillance region of low to high LSI.

It can be observed from Figure 4.19 that a relatively high value of $P > 0.7$ is reached during tracking. However, there is no further flight motion that indicates an impending attack, which would have caused an increase in P. In this situation, $P' > P$, with $P' \in (0.8, 0.85)$ attained. In view of the high values for P and P' , it is very likely for the defenders to be on high alert against possible attack by the aircraft.

Example 3b: Aircraft cruising in surveillance region of low LSI.

This example is analogous to Example 3a, with the entire surveillance region being of low LSI. It is apparent from Figure 4.20 that the values of P obtained are almost identical to those obtained in Example 3a. Due to the low LSI of the surveillance region, P' remains at a lower level of about 0.5 throughout the approach. It appears from the simulation results that there is no immediate need to raise the defenders' alert against

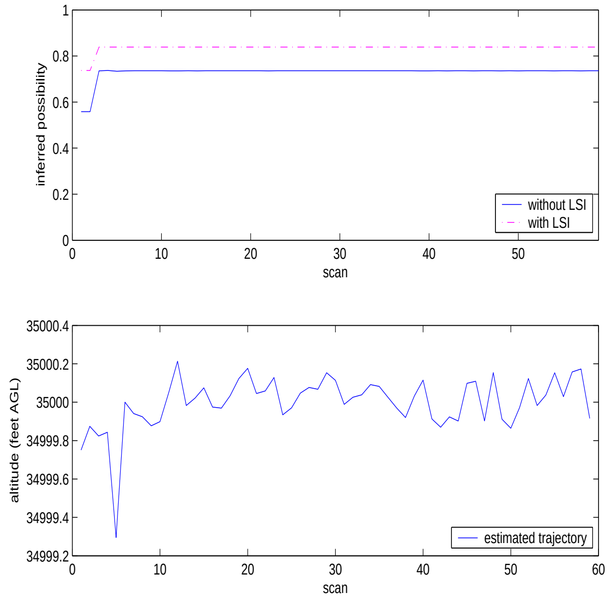

Figure 4.19: Example 3a - Fuzzy inference system output.

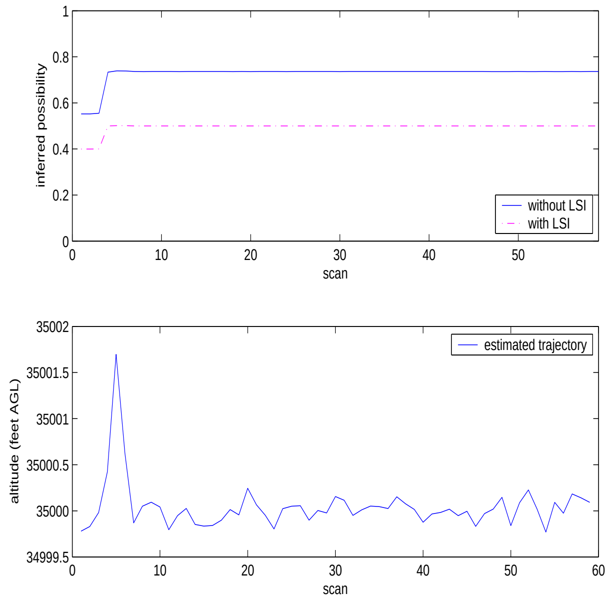

Figure 4.20: Example 3b - Fuzzy inference system output.

the aircraft.

Example 4: Aircraft unlikely to launch an attack.

Figure 4.21 shows an instance of results obtained for the simulated flight trajectory of an aircraft which is unlikely to carry out a weapon delivery, such as one that is performing aerobatics. It can be seen that P, as well as P', is always below the proposed threshold value of 0.85 for classifying an aircraft as having adversarial intent.

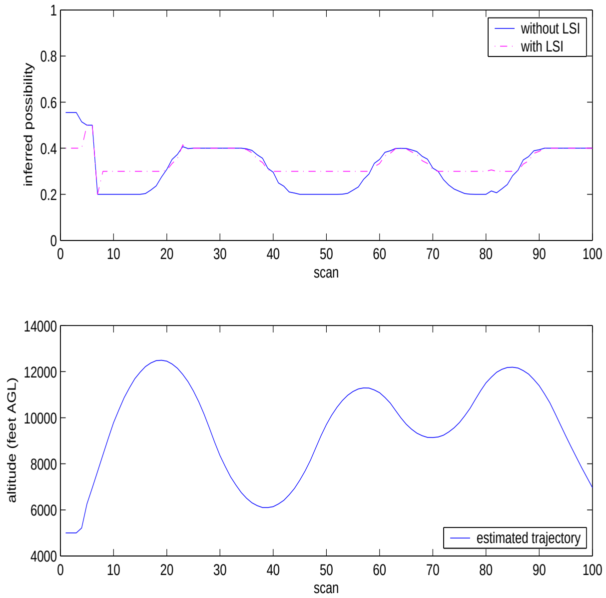

Figure 4.21: Example 4 - Fuzzy inference system output.

4.5.2 Conformance Monitoring

Consider the planned flight trajectory shown in Figure 4.22. Simulation tests are carried out on 100 flight profiles generated using different combinations of flight parameters (based on existing computation formulas and constraints). For each test, Procedure 2 (see Section 4.4.1) is carried out to obtain the inferred possibility of non-conformance in the behaviour of the tracked aircraft. We categorize aircraft behaviour into three types, namely, *conforming*, *non-conforming* and *ambiguous* [269], in our discussion. Figure 4.23 depicts typical simulation results obtained.

For the conforming case, FIS output values (denoted by P'' henceforth) remain consistently moderate throughout the tracking process. The corresponding deviations from planed states (namely, position, velocity and heading) are relatively small. For the non-conforming case, P'' rises rapidly after an initial period of low to moderate values during tracking. The surge in P'' is due to significant increases in state deviations. The third type of aircraft behaviour is considered ambiguous due to indefiniteness in the behavioural traits represented by P''. In this case, there exist instances when P'' increases to become sufficiently large to indicate non-conformance, where corresponding state deviations manifest aberrant behaviour in aircraft manœuvre. However, P'' subsequently decreases to the extent that conformance is signified, where corresponding state deviations provide evidence of a shift towards the right direction of travel.

It appears from the simulation results that aircraft behaviour can be deemed non-

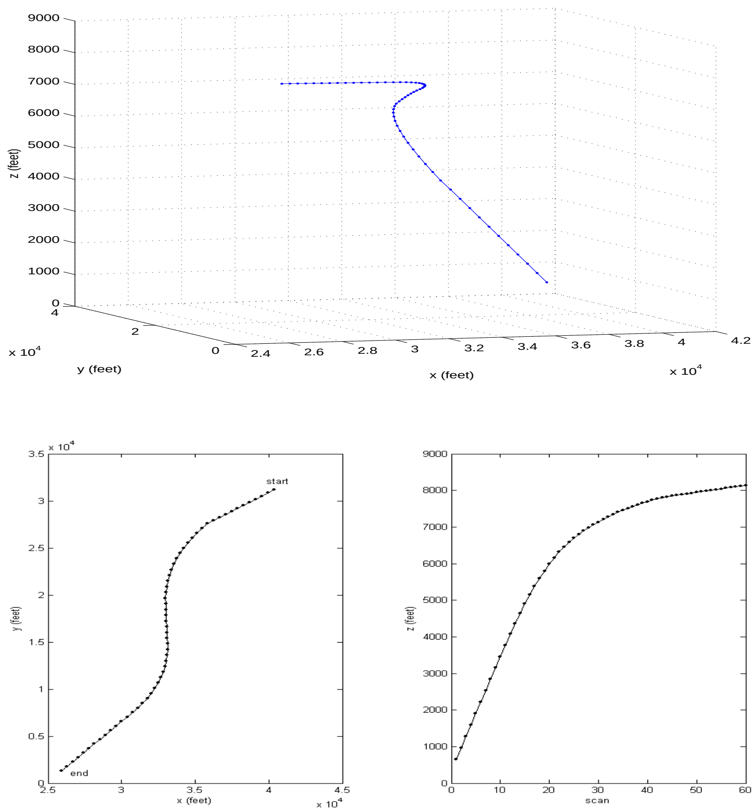

Figure 4.22: Planned flight trajectory.

conforming when $P'' > 0.85$. It is suggested that alert against non-conformance should be raised when $P'' > 0.7$. This would enable ATC/ATM system controllers to provide the pilot with early warning against navigating beyond safety limits. Consequently, the pilot would likely be able to execute necessary manœuvres to steer back towards the planned trajectory with less delay.

4.6 Comparison of Algorithms for State Estimation

Consider the problem on inferring the intent of the pilot of an aircraft being tracked by a military surveillance system, which was discussed in Section 4.3.2. In the proposed procedure for inferring the possibility of weapon delivery by a tracked attack aircraft, the IMM algorithm (using EKF in every model) is applied for the task of kinematic state estimation (Step 2 of Procedure 1 in Section 4.3.2).

Here, simulation tests are carried out in a way similar to that described in Sections 4.5.1. Four typical flight profiles for offset pop-up delivery (see Figure 4.2), generated with different combinations of flight parameters, are used as target trajectories in the tests. IMM algorithm variants from Table 3.1 which use either regularized, standard,

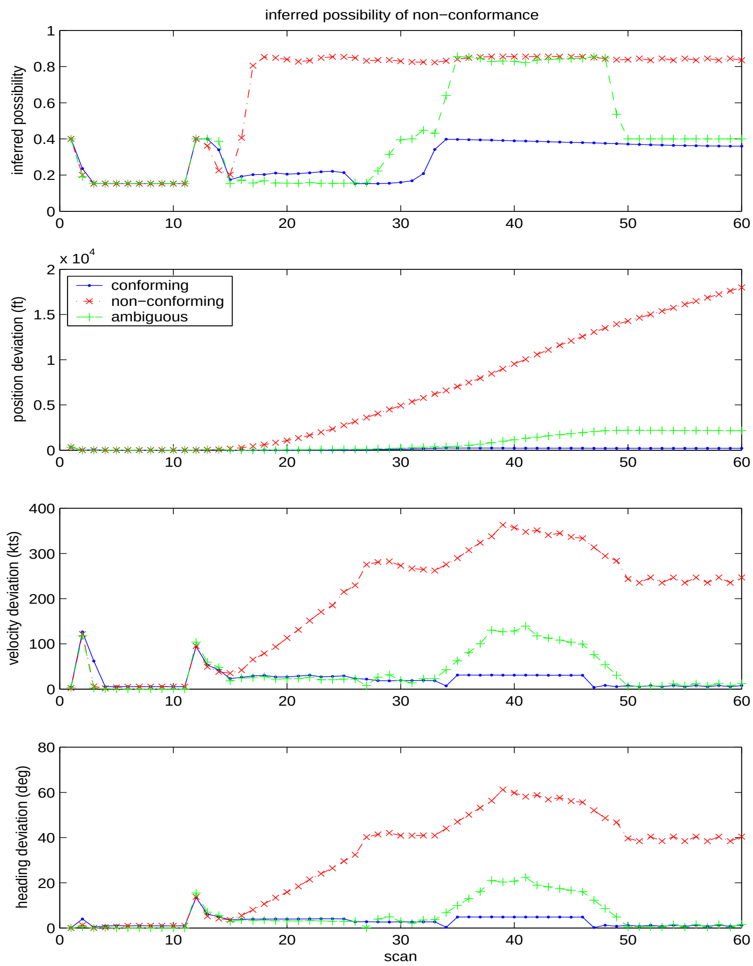

Figure 4.23: Fuzzy inference system output (conformance monitoring).

auxiliary or Gaussian particle filters are implemented for kinematic state estimation at Step 2 of Procedure 1 in Section 4.3.2. The system models are as given in Section 3.4.1. The correlation/covariance matrices for the process noise and the measurement noise are both set to scalar matrices. The sampling interval is $T = 1$ (second). For each particle filter used, the number of particles is $N_s = 100$. The total number of independent simulation runs is $C = 100$.

4.6.1 Numerical Results

The IMM algorithm variant implemented in Section 4.5.1, IEK, is used as the basis case for comparison of the performance of the algorithms implemented.

4.6.1.1 Computational Complexity

As done in Sections 3.4.1.2, first consider the processing time (in seconds). Let t_X (respectively, $\tilde{t}_X$) denote the mean processing time per simulation run (respectively, per scan) required by Algorithm X. Define τ_X (respectively, $\tilde{\tau}_X$) as the average of t_X (respectively, $\tilde{t}_X$) calculated over the four sets of simulation tests. To measure the relative computational complexity of X with respect to IEK, the ratio τ_X/τ_{IEK} (respectively, $\tilde{\tau}_X/\tilde{\tau}_{IEK}$) is computed. Then express the number of operations required for one cycle of each algorithm using the big-O notation (in terms of the number of models $r = 3$, state dimension $n_x = 9$ and $N_s = 100$). Tables 4.5 and 4.6 show the computational complexity of each IMM algorithm variant. The last column displays the relative computational complexity of each variant with respect to IEK.

As done earlier in Chapter 3, consider the relationship between the mean processing time per simulation run and the analytic time complexity for each filtering algorithm X implemented. Let $O(b_X)$ denote the analytic time order for algorithm X, where $b(\cdot)$ is a function of r, n_x and N_s. We use the definitions from Equations 3.41 and 3.42 to get

$$\hat{r}_X := \frac{\tau_X}{b_X},$$

and the corresponding normalization with respect to $\hat{r}_{IEK}$,

$$r_X := \frac{\hat{r}_X}{\hat{r}_{IEK}}.$$

It is shown in Figure 4.24 that for each filtering algorithm X implemented, r_X is a positive constant of moderate magnitude. It can be inferred from the results that the processing time for each filtering algorithm is proportional to its computational complexity (in terms of the corresponding analytic time order). As mentioned in Section 3.4.1.2, the relative processing time for any two arbitrary filtering algorithms is proportional to their relative computational complexity (in terms of analytic time order).

It is important for the military defenders to be able to achieve timely inference, as well as to have sufficient time for reaction to the motion of the tracked aircraft. The ideal situation is to be able to minimize the amount of processing time without the accuracy of results being compromised. It can be seen from Tables 4.5 and 4.6 that the mean processing time per scan required by each implemented algorithm is less than the sampling interval $T = 1$ (second), ranging from less than 0.015 seconds for IEK and IUK to approximately 0.16 seconds for algorithms that use auxiliary particle filters. Thus, for each algorithm implemented, there would be time for reaction to the target motion before the next scan. On the other hand, IMM algorithms which use extended Kalman particle filters and unscented particle filters are not considered for the current problem due to practical reasons. In particular, implementation of each of these algorithms is

Filter X	Big-O notation	Mean processing time per simulation run (s)					
		Trajectory 1	Trajectory 2	Trajectory 3	Trajectory 4	Average	τ_X/τ_{IEK}
IEK	$3n_x^3$	0.484	0.639	0.795	0.661	0.645	1.00
IUK	$3n_x^3$	0.202	0.256	0.340	0.330	0.282	0.44
IEG	$N_s n_x^2$	4.569	5.837	7.334	7.550	6.322	9.80
IER	$N_s n_x^2$	2.591	3.653	4.460	4.610	3.828	5.94
IES	$N_s n_x^2$	2.591	3.622	4.093	4.508	3.704	5.74
IEA	$N_s n_x^2$	5.573	7.059	8.899	9.021	7.638	11.84
IUG	$N_s n_x^2$	4.516	5.586	7.135	7.231	6.117	9.48
IUR	$N_s n_x^2$	2.618	3.344	4.306	4.350	3.655	5.67
IUS	$N_s n_x^2$	2.614	3.338	4.136	4.347	3.609	5.60
IUA	$N_s n_x^2$	5.541	6.809	8.768	8.880	7.500	11.63
IEUG	$N_s n_x^2$	4.514	5.773	7.122	7.222	6.158	9.55
IEUR	$N_s n_x^2$	2.615	3.352	4.301	4.559	3.707	5.75
IEUS	$N_s n_x^2$	2.614	3.500	4.135	4.345	3.648	5.66
IEUA	$N_s n_x^2$	5.595	6.954	8.767	9.071	7.597	11.78

Table 4.5: Computational complexity (per simulation run).

Filter X	Big-O notation	Mean processing time per scan (s)					
		Trajectory 1	Trajectory 2	Trajectory 3	Trajectory 4	Average	$\tilde{\tau}_X/\tilde{\tau}_{IEK}$
IEK	$3n_x^3$	0.014	0.015	0.014	0.012	0.014	1.00
IUK	$3n_x^3$	0.006	0.006	0.006	0.006	0.006	0.43
IEG	$N_s n_x^2$	0.131	0.133	0.133	0.135	0.133	9.73
IER	$N_s n_x^2$	0.074	0.083	0.081	0.082	0.080	5.87
IES	$N_s n_x^2$	0.074	0.082	0.074	0.080	0.078	5.70
IEA	$N_s n_x^2$	0.159	0.160	0.162	0.161	0.161	11.76
IUG	$N_s n_x^2$	0.129	0.127	0.130	0.129	0.129	9.43
IUR	$N_s n_x^2$	0.075	0.076	0.078	0.078	0.077	5.62
IUS	$N_s n_x^2$	0.075	0.076	0.075	0.078	0.076	5.55
IUA	$N_s n_x^2$	0.158	0.155	0.159	0.159	0.158	11.55
IEUG	$N_s n_x^2$	0.129	0.131	0.129	0.129	0.130	9.49
IEUR	$N_s n_x^2$	0.075	0.076	0.078	0.081	0.078	5.68
IEUS	$N_s n_x^2$	0.075	0.080	0.075	0.078	0.077	5.62
IEUA	$N_s n_x^2$	0.160	0.158	0.159	0.162	0.160	11.70

Table 4.6: Computational complexity (per scan).

likely to incur comparatively large computation time (close to T) that may result in insufficient response time for the decision makers.

When a particle filter is used, the amount of processing time required is directly proportional to the number of samples used. In such a case, one important objective is to try to achieve satisfactory results without the need to increase the number of samples.

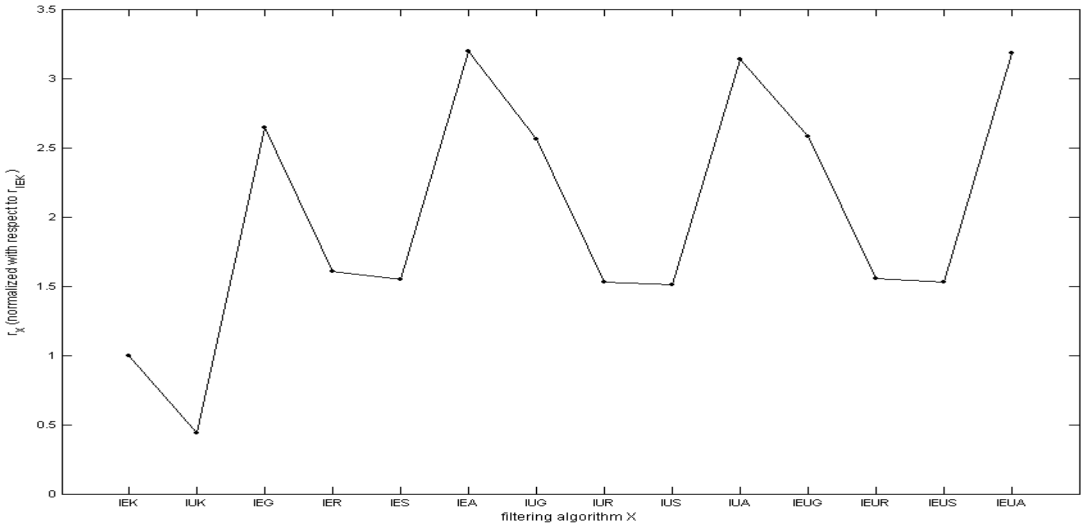

Figure 4.24: Processing time relative to analytic time complexity.

4.6.1.2 Analysis of Results

Analysis of the simulation test results is done as follows:

1. comparison of state estimation errors obtained with the different filtering algorithms;
2. statistical comparison of the filtering algorithms, formulated as a hypothesis problem.

1. Comparison of estimation errors

Let L denote the total number of scans for the duration of tracking a target (that is, the number of points on a target trajectory). The RMSEs in estimating the possibility of weapon delivery by the tracked aircraft is defined as

$$\text{RMSE} := \left\{ \frac{1}{C} \frac{1}{L} \sum_{i=1}^{C} \sum_{k=1}^{L} |\hat{u}_{ik} - u_k|^2 \right\}^{\frac{1}{2}},$$

where

- $\hat{u}_{ik}$ is the inferred possibility (fuzzy inference system output) obtained with the filtered flight trajectory at the k-th scan in the i-th simulation run, $i = 1, \ldots, C$, $k = 1, \ldots, L$, and
- u_k is the inferred possibility that would be obtained with the actual flight trajectory at the k-th scan, $k = 1, \ldots, L$.

Two cases of estimation of the possibility of weapon delivery are investigated: firstly, without the location sensitivity index (see Sections 4.3.2.1 and 4.5.1) considered and secondly, with the LSI considered. In the discussion below, the first and the second

Filter X	RMSE (to 3 decimal places)					
	Trajectory 1	Trajectory 2	Trajectory 3	Trajectory 4	Average	φ_X/φ_{IEK}
IEK	0.309	0.347	0.312	0.279	0.312	1.00
IUK	0.152	0.159	0.153	0.156	0.155	0.50
IEG	0.167	0.176	0.170	0.194	0.177	0.57
IER	0.136	0.174	0.144	0.149	0.151	0.48
IES	0.137	0.169	0.122	0.143	0.143	0.46
IEA	0.146	0.173	0.145	0.142	0.152	0.49
IUG	0.147	0.147	0.138	0.170	0.151	0.48
IUR	0.139	0.128	0.128	0.124	0.130	0.42
IUS	0.142	0.135	0.122	0.120	0.130	0.42
IUA	0.145	0.130	0.125	0.119	0.130	0.42
IEUG	0.151	0.157	0.141	0.172	0.155	0.50
IEUR	0.141	0.138	0.130	0.136	0.136	0.44
IEUS	0.141	0.149	0.124	0.123	0.134	0.43
IEUA	0.150	0.150	0.123	0.139	0.140	0.45

Table 4.7: Errors in estimating the inferred possibility of weapon delivery (without LSI).

cases are referred to as Case 1 and Case 2 respectively. Tables 4.7 and 4.8 show the RMSEs in estimating the possibility of weapon delivery in Cases 1 and 2 respectively. For each filtering algorithm, the average RMSEs in estimating the possibility of weapon delivery, computed over the four sets of simulation, are also tabulated. In the tables, φ_X denotes the average RMSE in estimating the possibility of weapon delivery with algorithm X used for kinematic state estimation.

It can be seen from the tables that all the implemented algorithms yield smaller estimation errors than IEK. In Case 1, IUR and IUS attain average RMSEs that are about 40% of that obtained with IEK. Each of IUR and IUS has a computational load that is larger than those of IEK and IUK (less than 6 times that of IEK). The tremendous increase in computational load is incurred when a particle filter is used in place of EKF/UKF in the coordinated turn model. However, it is still modest compared to the algorithms that use GPF or APF. IUA yields comparable results, but requires about twice the computational load required by an IMM algorithm that uses RPF or SPF. The estimation results obtained with the remaining IMM algorithms are worse, in terms of larger estimation errors (about 50–60% of that obtained with IEK). It is noted that the average RMSE obtained with IUK is about 50% of that obtained with IEK, but IUK requires lower computational cost than the IMM algorithms which use particle filters (at least 12 times less), and IEK. The situation is similar in Case 2, with average RMSEs obtained with IUR, IUS and IUA being about 20% of that obtained with IEK.

Filter X	RMSE (to 3 decimal places)					
	Trajectory 1	Trajectory 2	Trajectory 3	Trajectory 4	Average	φ_X/φ_{IEK}
IEK	0.302	0.370	0.344	0.295	0.328	1.00
IUK	0.078	0.090	0.101	0.089	0.090	0.27
IEG	0.098	0.133	0.117	0.136	0.121	0.37
IER	0.066	0.148	0.108	0.119	0.111	0.34
IES	0.066	0.141	0.068	0.106	0.095	0.29
IEA	0.085	0.141	0.110	0.105	0.110	0.34
IUG	0.076	0.082	0.078	0.092	0.082	0.25
IUR	0.066	0.071	0.072	0.070	0.070	0.21
IUS	0.070	0.075	0.069	0.067	0.070	0.21
IUA	0.071	0.073	0.069	0.065	0.070	0.21
IEUG	0.078	0.114	0.076	0.093	0.090	0.27
IEUR	0.069	0.084	0.074	0.104	0.083	0.25
IEUS	0.070	0.111	0.069	0.067	0.079	0.24
IEUA	0.087	0.112	0.069	0.105	0.093	0.28

Table 4.8: Errors in estimating the possibility of weapon delivery (with LSI).

The remaining IMM algorithms which use particle filters yield average RMSEs of about 25–40% of that obtained with IEK. In addition, the average RMSEs obtained with IUK is about 30% of that obtained with IEK.

2. Statistical comparison of filtering algorithms

As in Section 3.4.1.3, comparison of the performance (in terms of the mean square error in the estimation of the possibility of weapon delivery) of the filtering algorithms implemented is formulated as a hypothesis testing problem as follows [13, Sections 1.5 and 11.5].

Let X be an arbitrary filtering algorithm implemented in the preceding simulation tests. Let J_X denote the actual mean square error in estimation. The hypothesis testing problem is to test the null hypothesis

$$H_0 : \Delta = J_{IEK} - J_X \leq 0 \quad \text{(algorithm } X \text{ not better than IEK)}, \tag{4.3}$$

versus the alternate hypothesis

$$H_1 : \Delta = J_{IEK} - J_X > 0 \quad \text{(algorithm } X \text{ better than IEK)}, \tag{4.4}$$

subject to

$$\text{Prob}\{\text{accept } H_1 | H_0 \text{ is true}\} = \alpha \quad \text{(level of significance for hypothesis } H_0\text{)}.$$

With reference to Equation 3.48, the statistical test is based on the independent mean square error differences,

$$(\Delta_X)_i := \frac{1}{L}\sum_{k=1}^{L}\Big\{\big|(\hat{u}_{IEK})_{ik} - u_k\big|_2^2 - \big|(\hat{u}_X)_{ik} - u_k\big|_2^2\Big\}, \quad i = 1, \ldots, C, \tag{4.5}$$

where

- $(\hat{u}_X)_{ik}$ is the inferred possibility of weapon delivery obtained with the filtered flight trajectory (without LSI) for algorithm X at the k-th scan in the i-th simulation run, $i = 1, \ldots, C$, $k = 1, \ldots, L$, and
- u_k is the inferred possibility that would be obtained with the actual flight trajectory at the k-th scan, $k = 1, \ldots, L$.

As in Section 3.4.1.3, we use $\alpha = 0.05$ (equivalently, 5% level of significance). Thus, H_1 is accepted if the test statistic

$$\rho = \frac{\bar{\Delta}_X}{\sigma_{\bar{\Delta}_X}} > 1.65.$$

Tables 4.9 and 4.10 show the test for differences of mean square errors in estimating the possibility of weapon delivery, between IEK and each of the other filtering algorithms. It can be seen from the tabulated results that all the algorithms implemented have statistically significant improvement over IEK (that is, test statistic $\rho > 1.65$) in the estimation of the possibility of weapon delivery.

Based on the above discussion, as well as the constraints on computation time requirements, it is probably reasonable to choose IUR and IUS over the remaining algorithms in practical implementation. This would incur relatively modest computational cost, as well as minimal or no loss in the accuracy of results. On the other hand, in circumstances with the level of accuracy in results being less crucial than computational load, the preferred choice of algorithm would probably be IUK. IUK yields average RMSEs of about 50% and 30% of that obtained with IEK in Case 1 and Case 2 respectively. However, IUK and IEK have comparable computational load. Compared to the previously mentioned algorithms with better results in state estimation, using IUK can yield much saving in computational costs/time.

4.7 Approach by More than One Aircraft

Our proposed method deals with intent inference for a single aircraft. The problem on handling an approach by multiple aircraft in military surveillance and air traffic

Filter X	Test statistic for estimation of the possibility of weapon delivery			
	Trajectory 1	Trajectory 2	Trajectory 3	Trajectory 4
IUK	6.42	7.85	6.50	6.01
IEG	5.90	7.06	6.28	4.66
IER	6.84	7.02	7.13	6.52
IES	6.81	7.65	7.51	6.77
IEA	6.48	7.67	7.02	6.71
IUG	6.56	8.18	7.11	5.44
IUR	6.80	8.52	7.41	7.13
IUS	6.64	8.48	7.53	7.08
IUA	6.55	8.58	7.44	7.15
IEUG	6.44	7.99	7.03	5.47
IEUR	6.66	8.29	7.39	6.55
IEUS	6.65	8.24	7.43	7.05
IEUA	6.45	8.13	7.50	6.75

Table 4.9: Comparison of IEK with other IMM variants in estimation of the possibility of weapon delivery (without LSI).

Filter X	Test statistic for estimation of the possibility of weapon delivery			
	Trajectory 1	Trajectory 2	Trajectory 3	Trajectory 4
IUK	6.58	7.96	6.65	6.10
IEG	6.14	7.14	6.80	5.56
IER	6.71	6.75	6.94	5.89
IES	6.70	7.45	7.26	6.06
IEA	6.31	7.48	6.90	6.06
IUG	6.61	8.06	7.15	6.06
IUR	6.71	8.15	7.24	6.36
IUS	6.66	8.14	7.25	6.37
IUA	6.64	8.16	7.25	6.38
IEUG	6.55	7.78	7.17	6.07
IEUR	6.67	8.00	7.21	5.78
IEUS	6.64	7.83	7.25	6.37
IEUA	6.30	7.80	7.25	6.05

Table 4.10: Comparison of IEK with other IMM variants in estimation of the possibility of weapon delivery (with LSI).

control/management would be more complex and would require much additional consideration. Some issues associated with this problem are discussed below.

4.7.1 Flight Formation

The flight approach can be in individual form or in a formation. Some examples of flight formations employed by tactical combat aircraft are briefly described in this section [319].

4.7.1.1 Two-ship Formation

In a line abreast formation, the position of the wingman[1] relative to the flight leader is 0° to 20° aft, 4000 to 12000 feet spacing with altitude separation. A vertical stack of 2000 to 6000 feet is used, when applicable, to minimize the chance of simultaneous detection by an opponent.

For a fighting wing formation, the wingman is given a manœuvring cone from 30° to 70° aft of line abreast and lateral spacing between 500 and 3000 feet. This formation is employed when maximum manœuvring potential is desired.

4.7.1.2 Four-ship Formation

The four-ship formation is employed under the control of one flight leader. It is employed as a single entity as long as it is not forced to separate into a lead element (flight leader and his wingman) and a second/trailing element (second leader and his wingman).

In a box formation, the two-ship elements use basic line abreast manœuvring and principles concerning lookout responsibilities. Depending on terrain and weather, the trailing element takes 1.5 to 3 nautical miles separation from the lead element. The spacing serves the purpose of maximizing separation to avoid easy visual detection of the entire flight formation. Manœuvres are initiated by the element leaders in this formation.

For a fluid four formation, the element leaders maintain line abreast formation, while their wingmen assume fighting wing. The flight leader is at the front of the formation, with his wingman to his rear left. The second leader is to the rear right of the flight leader, while his wingman assume fighting wing. The assembly of four of these formations forms a squadron formation.

In a spread four formation, the element leaders maintain the same spacing as for the fluid four formation. The wingmen position themselves 0° to 30° to the rear of their respective element leaders at 6000 to 9000 feet spread. The increase in lateral spacing for wingmen facilitates manœuvring. The elements need not always be line abreast. There may be instances when are briefly in trail. Spread formation makes it difficult to visually acquire the entire flight formation.

A three-ship contingency formation can be considered as an alternative for a four-ship formation mission in some occasions. It is obtained from the four-ship formation concerned by having an appropriate flight member fall out from the original formation.

[1]Wingman: in a formation of aircraft, the pilot who flies behind and to the side of the leader.

4.7.1.3 Echelon Formation

The flight members are arranged diagonally in an echelon formation. Each member is positioned to the rear right, or to the rear left, of the member ahead. These two types of formations are known as a right echelon and a left echelon respectively.

4.7.2 Multiple Target Tracking and Identity Management

The problem of dealing with approach by more than one aircraft requires the employment of multiple target tracking techniques [33,34,58,145,147,243,261] for the state estimation component of our proposed intent inference method. For each tracked aircraft, information based on the estimated kinematic states need to be taken into consideration for processing by a fuzzy inference system, in order to derive the pilot intent.

As mentioned before, the amount of computational load/time is a critical factor for the two intent inference problems discussed here. Hence, it is desirable to select multiple target tracking algorithms with modest time complexities.

Another point of concern is the detection and identification of the targets under surveillance. It may be difficult to distinguish the targets from one another during tracking when there is close proximity and/or interaction among them, such as in the case of a tactical aircraft formation.

To address the aforementioned issues, the multiple-target tracking and identity management (MTIM) algorithm developed in [147] could be considered. The MTIM algorithm is constituted of the following components:

- data association - uses a computationally efficient algorithm based on the joint probabilistic data association algorithm [230], in which measurement data is associated with targets via the use of target kinematic information (position and velocity);

- tracking/hybrid state estimation - uses residual-mean IMM algorithm based on multiple aircraft dynamics models; and

- identity management - uses an algorithm with the ability to keep track of target identities via the use of local attribute information about them (either explicitly available from sensors or inferred from a technique based on the multiple hypothesis tracking algorithm [230]).

The applicability of the MTIM algorithm for incorporation into the intent inference method proposed in this chapter could be investigated as part of our future research.

4.8 Summary

In this chapter, we have presented an approach for intent inference, which concerns the use of available knowledge on the preceding activities of a target of interest to predict its future action. The approach is based on the analysis of aircraft flight profiles. The method is implemented for two applications.

Firstly, it has been shown that it is possible to infer the intent of an attack aircraft, particularly on its weapon delivery. The proposed approach is extended to consider the environmental context of the tracked aircraft when executing the inference process. Simulation is carried out on four test examples with different scenarios to evaluate the performance of the method. The results verify the feasibility of the method and its ability to provide timely inference. It is also justifiable to consider the environmental context, which is useful in raising military defenders' level of vigilance early against potential adversaries, hence allowing more time to prepare for pre-emptive action.

In the second application, experimental results show that the proposed solution has much potential in being a useful tool for conformance monitoring in ATC/ATM. It can be used to assist ATC/ATM system controllers in determining whether aircraft are deviating from or adhering to designated courses of travel. As a result, corrective/remedial actions can be taken once deviant behaviour is detected.

Next, IMM algorithm variants developed in Chapter 3 have been implemented for the state estimation component of the proposed intent inference algorithm. Based on the problem on intent inference for an attack aircraft which was discussed in Section 4.3, the effectiveness and efficiency of the algorithms are compared and analyzed.

Our proposed intent inference method has only considered approach by a single aircraft. We briefly discuss the extension of the proposed method to deal with an approach by multiple aircraft, such as that by a flight formation. Some of the main issues concerned include multiple target tracking and management of the target identities. These topics are of interest in our future research.

Chapter 5

Conclusion and Further Research

Data and information fusion (DIF) is a multidisciplinary research field that strides across Computing, Engineering and Science. Subjects/techniques involved include, but are not limited to, Artificial Intelligence, Control Theory, Cognitive Psychology, Computational Intelligence, Information Theory, Mathematical Logic, Signal Processing, Software Engineering, as well as Probability and Statistics [39, 230].

This chapter provides a summary of research work on some issues and applications of DIF done in this thesis. Possible areas for further research are also discussed.

5.1 Summary

Chapter 2 contains a survey of high-level information fusion. It presents a review of several DIF models and existing work on high-level information fusion. A discussion on some common application areas of DIF is included. Topics with potential for further research are also mentioned.

Chapter 3 discusses the proposal of an interacting multiple model based approach for target tracking. Various combinations of extended Kalman filters, unscented Kalman filters and particle filters are used for the models. Each IMM algorithm variant uses a constant velocity model, a constant acceleration model and a coordinated turn model. The filtering algorithms are tested on two problems. The first problem concerns three-dimensional manœuvring target tracking, while the second deals with localization and tracking in a horizontal plane with the use of a time difference of arrival system. Simulation test results indicate that IMM algorithms which use regularized or standard particle filters in the coordinated turn model, with unscented Kalman filters in the remaining two models, perform better than the other filters, in terms of smaller state estimation errors obtained. The filtering algorithms are also implemented for a problem on modelling the prices of financial option contracts, which uses real data in the numerical tests.

Chapter 4 presents an approach for intent inference based on the analysis of flight profiles. The method is tested on two applications. The first one is military surveillance of attack aircraft. Four different scenarios are considered to assess the practicability of the suggested method. Another application is conformance monitoring in air traffic control/management systems. Results obtained from simulation tests demonstrate that the proposed method shows promise in being a useful tool to assist critical human decision making. For the application problem on military surveillance, several IMM algorithm variants discussed in Chapter 3 are also tested for viability in the state estimation component of the proposed intent inference approach. In particular, constraints on computation time requirements are taken into consideration when deciding on the algorithm variants to be used.

5.2 Further Research

With reference to the research done in this thesis, topics for further investigation are described below.

5.2.1 Target Tracking

Further investigation on the manœuvring target tracking problem can be done by modelling the measurement/observation noise as glint coloured noise instead of white noise [215,329–331]. Many different models can also be considered for handling manœuvres [191–196].

The current work on single target tracking can be extended to the more general case of multiple target tracking [12,21,33,34,58,145,147,180,243,261,305].

There are many more nonlinear filters that can be considered for state estimation. Examples include the Gauss-Hermite filter [68,150], the adaptive unscented particle filters (AUPF, AUPF2) [68], the multiple-model based AUPFs (MM-AUPF, MM-AUPF2) [68], the marginalized particle filter [292], the Gauss-Newton particle filter [48], as well as the probability hypothesis density filter (PHDF) [211]. It is also useful to investigate the combination of different filters for effective and efficient multiple model based state estimation.

When implementing a particle filter, one possible issue of concern is the effect of varying N_s (the number of samples) and N_T (the threshold for the effective sample size

N_{eff}) in the filter. The aim would be to achieve similar or better performance with smaller N_s and/or N_T, hence saving on computational costs.

We realize from the numerical results in Chapter 3 (Sections 3.4.1.3 and 3.4.2.3) that the filtering algorithms are not totally consistent, that is, not all state estimation errors are commensurate with estimator-calculated covariances. It is acknowledged that it is not possible for adaptive algorithms to be entirely consistent [12], as timely adaptation is driven by the inconsistency. However, it is important in practical implementation to strive to attain consistency and also achieve low estimation errors [13]. This would require investigation on the topic of filter tuning.

The filtering algorithms discussed in Chapter 3 have been implemented for a real world problem on pricing financial options. It would be useful to carry out more tests on other relevant applications for performance evaluation of the algorithms.

5.2.2 Intent Inference

Further development and improvement on the intent inference approach proposed in Chapter 4 can be done. In the event that additional information on the target of interest is available, it would probably be possible to introduce more input variables and fuzzy rules to the fuzzy inference system. This is likely to enhance the performance of the fuzzy inference system and hence, yield desired system output of higher quality.

There are many inference mechanisms that can be explored for the inference process. Some instances are Bayesian networks, Hybrid inference (a technique that combines Bayesian networks and fuzzy logic) [190] and other possible combinations of techniques with complementary properties (for example, fuzzy logic and neural networks [1]).

Our proposed intent inference method currently considers intent inference for single aircraft. It would be useful to extend it to deal with the more realistic situation of handling multiple aircraft. Some of the main issues concerned include computation time requirements (due to the need for fast and accurate response against opponents), multiple target tracking and management of the target identities (briefly discussed in Section 4.7). These topics would need to be taken into consideration for further development and enhancement of our current intent inference system.

Bibliography

[1] A. Abraham, Adaptation of fuzzy inference system using neural learning, Studies in Fuzziness and Soft Computing, 181:53-83, 2005.

[2] S. Ahlberg, P. Hörling, K. Johansson, K. Jöred, H. Kjellström, C. Mårtenson, G. Neider, J. Schubert, P. Svenson, P. Svensson, J. Walter, An information fusion demonstrator for tactical intelligence processing in network-based defense, Information Fusion, 8(1):84–107, 2007.

[3] T. Aldinger, J. Kao, Data fusion and theater undersea warfare - an oceanographer's perspective, in: Proceedings of the MTS/IEEE Oceans Conference, 4:2008–2012, 2004.

[4] H. Alex, M. Kumar, B. Shirazi, MidFusion: an adaptive middleware for information fusion in sensor network applications, Information Fusion, 9(3):332–343, 2008.

[5] A. Aligawesa, K. L. Fetzer, Intent Inference and Conformance Monitoring in Air Traffic Control, Project Report, Flight Dynamics and Control/Hybrid Systems Laboratory, School of Aeronautics and Astronautics, Purdue University, 2008.

[6] M. K. Allouche, Real-time use of Kohonen's self-organizing maps for threat stabilization, Information Fusion, 6(2):153–163, 2005.

[7] X. D. An, G. Q. Liu, Application of neural network in the field of target threat evaluation, in: Proceedings of the IEEE International Joint Conference on Neural Networks, 6:4237–4240, 1999.

[8] C. Anken, N. Gemelli, P. LaMonica, R. Mineo, J. Spina, Intelligent systems technology for higher level fusion, in: Proceedings of the International Conference on Information Fusion, 2:1316–1323, 2002.

[9] S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Transactions on Signal Processing, 50(2):174–188, 2002.

[10] J. M. Aughenbaugh, B. R. La Cour, Measurement prioritization for optimal Bayesian fusion, in: Proceedings of the International Conference on Information Fusion, Paper 1423, 2007.

[11] K. Baclawski, M. Kokar, J. Letkowski, C. Matheus, M. Malczewski, Formalization of situation awareness, in: Proceedings of the OOPSLA Workshop on Behavioral Semantics, pp. 1–15, 2002.

[12] Y. Bar-Shalom, X.-R. Li, Multitarget-Multisensor Tracking: Principles and Techniques, YBS Publishing, ISBN: 0964831201, 1995.

[13] Y. Bar-Shalom, X.-R. Li, T. Kirubarajan, Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software, John Wiley & Sons, ISBN: 047141655X, 2001.

[14] O. Basir, X. Yuan, Engine fault diagnosis based on multi-sensor information fusion using Dempster-Shafer evidence theory, Information Fusion, 8(4):379–386, 2007.

[15] M. Bedworth, J. O'Brien, The Omnibus model: a new model of data fusion?, IEEE Aerospace and Electronic Systems Magazine, 15(4):30–36, 2000.

[16] B. Bell, E. Santos Jr., S. M. Brown, Making adversary decision modeling tractable with intent inference and information fusion, in: Proceedings of the Conference on Computer Generated Forces and Behavioral Representation, pp. 535–542, 2002.

[17] D. Bellot, A. Boyer, F. Charpillet, A new definition of qualified gain in a data fusion process: application to telemedicine, in: Proceedings of the International Conference on Information Fusion, 2:865–872, 2002.

[18] A. R. Benaskeur, H. Irandoust, P. Mcguire, R. Brennan, Holonic control-based sensor management, in: Proceedings of the International Conference on Information Fusion, Paper 1132, 2007.

[19] A. Benavoli, B. Ristic, A. Farina, M. Oxenham, L. Chisci, An approach to threat assessment based on evidential networks, in: Proceedings of the International Conference on Information Fusion, Paper 1213, 2007.

[20] A. Berrached, M. Beheshti, A. de Korvin, R. Alo, Applying fuzzy relation equations to threat analysis, in: Proceedings of the Annual Hawaii International Conference on System Sciences, pp. 684–688, 2002.

[21] S. Blackman, Multiple-Target Tracking with Radar Applications, Artech House, ISBN: 0890061793, 1986.

[22] S. S. Blackman, R. Popoli, Design and Analysis of Modern Tracking Systems, Artech House, ISBN: 1580530060, 1999.

[23] E. Blasch, Assembling a distributed fused information-based human-computer cognitive decision making tool, IEEE Aerospace and Electronic Systems Magazine, 15(5):11–17, 2000.

[24] E. Blasch, Level 5 (user refinement) issues supporting information fusion management, in: Proceedings of the International Conference on Information Fusion, Paper 130, 2006.

[25] E. Blasch, Sensor, user, mission (SUM) resource management and their interaction with level 2/3 fusion, in: Proceedings of the International Conference on Information Fusion, Paper 394, 2006.

[26] E. Blasch, I. Kadar, K. Hintz, J. Biermann, C. Chong, J. Salerno, S. Das, Resource management and its interaction with level 2/3 fusion - from the Fusion-2006 panel discussion, in: Proceedings of the International Conference on Information Fusion, Paper 1203, 2007.

[27] E. Blasch, I. Kadar, J. Salerno, M. M. Kokar, S. Das, G. M. Powell, D. D. Corkill, E. H. Ruspini, Issues and challenges in situation assessment (level 2 fusion), Journal of Advances in Information Fusion, 1(2):95–107, 2006.

[28] E. Blasch, S. Plano, JDL level 5 fusion model: user refinement issues and applications in group tracking, in: Proceedings of SPIE, 4729:270–279, 2002.

[29] E. Blasch, S. Plano, Level 5: user refinement to aid the fusion process, in: Proceedings of SPIE, 5099:288–297, 2003.

[30] E. Blasch, S. Plano, Ontological issues in higher levels of information fusion: user refinement of the fusion process, in: Proceedings of the International Conference on Information Fusion, 1:634–641, 2003.

[31] E. Blasch, S. Plano, DFIG level 5 (user refinement) issues supporting situational assessment reasoning, in: Proceedings of the International Conference on Information Fusion, 1:xxxv–xliii, 2005.

[32] H. A. P. Blom, E. A. Bloem, Particle filtering for stochastic hybrid systems, in: Proceedings of the IEEE Conference on Decision and Control, 3:3221–3226, 2004.

[33] H. A. P. Blom, E. A. Bloem, Joint particle filtering of multiple maneuvering targets from unassociated measurements, Journal of Advances in Information Fusion, 1(1):15–34, 2006.

[34] H. A. P. Blom, E. A. Bloem, Bayesian tracking of two possibly unresolved maneuvering targets, IEEE Transactions on Aerospace and Electronic Systems, 43(2):612–627, 2007.

[35] Y. Boers, J. N. Driessen, Interacting multiple model particle filter, IEE Proceedings - Radar, Sonar and Navigation, 150(5):344–349, 2003.

[36] N. A. Bomberger, B. J. Rhodes, M. Seibert, A. M. Waxman, Associative learning of vessel motion patterns for maritime situation awareness, in: Proceedings of the International Conference on Information Fusion, Paper 234, 2006.

[37] N. A. Bomberger, A. M. Waxman, B. J. Rhodes, N. A. Sheldon, A new approach to higher-level information fusion using associative learning in semantic networks of spiking neurons, Information Fusion, 8(3):227–251, 2007.

[38] É. Bossé, J. Roy, S. Paradis, Modeling and simulation in support of the design of a data fusion system, Information Fusion, 1(2):77–87, 2000.

[39] É. Bossé, J. Roy, S. Wark, Concepts, Models, and Tools for Information Fusion, Artech House, ISBN: 1596930810, 2007.

[40] É. Bossé, P. Valin, A.-C. Boury-Brisset, D. Grenier, Exploitation of a priori knowledge for information fusion, Information Fusion, 7(2):161–175, 2006.

[41] T. Bréhard, P.-A. Coquelin, E. Duflos, P. Vanheeghe, Optimal policies search for sensor management : application to the ESA radar, in: Proceedings of the International Conference on Information Fusion, Session SS04, 2008.

[42] B. Brehmer, The dynamic OODA loop: amalgamating Boyd's OODA loop and the cybernetic approach to command and control, in: Proceedings of the International Command and Control Research and Technology Symposium, 2005.

[43] R. Breton, R. Rousseau, THE C-OODA: A cognitive version of the OODA loop to represent C2 activities, in: Proceedings of the International Command and Control Research and Technology Symposium, 2005.

[44] R. Browne, C4I defensive infrastructure for survivability against multi-mode attacks, in: Proceedings of the 21st Century Military Communications Conference, 1:417–424, 2000.

[45] D. W. Burgess, C. L. Levins, Intelligent sensor resource management using evolutionary computing techniques, in: Proceedings of the International Conference on Integration of Knowledge Intensive Multi-Agent Systems, pp. 325–329, 2003.

[46] S. R. Byers, S. J. Yang, Real-time fusion and projection of network intrusion activity, in: Proceedings of the International Conference on Information Fusion, Session RS07, 2008.

[47] Y.-C. Cai, L. He, W.-M. Zhang, Z. Liu, A situation and threat assessment model based on group analysis, in: Proceedings of the International Conference on Machine Learning and Cybernetics, 1:356–361,2005.

[48] H. Cao, N. Ohnishi, Y. Takeuchi, T. Matsumoto, H. Kudo, Gauss-Newton particle filter, IEICE (The Institute of Electronics, Information and Communication Engineers) Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E90-A(6):1235–1239, 2007.

[49] G. T. Capraro, G. B. Berdan, J. Spina, R. A. Liuzzi, P. B. Berra, An architecture for providing information anytime, anywhere and on any device - an ontological approach, in: Proceedings of the International Conference on Information Fusion, 2:1331–1339, 2002.

[50] G. A. Carpenter, S. Martens, O. J. Ogas, Self-organizing information fusion and hierarchical knowledge discovery: a new framework using ARTMAP neural networks, Neural Networks, 18(3):287–295, 2005.

[51] H. S. Carvalho, W. B. Heinzelman, A. L. Murphy, C. J. N. Coelho, A general data fusion architecture, in: Proceedings of the International Conference on Information Fusion, 2:1465–1472, 2003.

[52] M. G. Ceruti, A. Ashenfelter, R. Brooks, G. Chen, S. Das, G. Raven, M. Sudit, E. Wright, Pedigree information for enhanced situation and threat assessment, in: Proceedings of the International Conference on Information Fusion, Paper 43, 2006.

[53] K. C. Chang, J. P. Hill, Level I and level II target valuations for sensor management, Journal of Advances in Information Fusion, 1(2):95–107, 2006.

[54] S. K. Chang, E. Jungert, X. Li, A progressive query language and interactive reasoner for information fusion support, Information Fusion, 8(1):70–83, 2007.

[55] A. I. Chao, E. Starczewski, B. C. Krikeles, A. E. Lusignan, An extensible, ontology-based, distributed information system architecture, in: Proceedings of the International Conference on Information Fusion, 1:642–649, 2003.

[56] A. R. Chaturvedi, M. Gupta, S. R. Mehta, W. T. Yue, Agent-based simulation approach to information warfare in the SEAS (Synthetic Environment for Analysis and Simulation) environment, in: Proceedings of the Annual Hawaii International Conference on System Sciences, vol. 1, 2000.

[57] Y. Chemin, K. Honda, Spatiotemporal fusion of rice actual evapotranspiration with genetic algorithms and an agrohydrological model, IEEE Transactions on Geoscience and Remote Sensing, 44(11, Part 2):3462–3469, 2006.

[58] B. Chen, J. K. Tugnait, Tracking of multiple maneuvering targets in clutter using IMM/JPDA filtering and fixed-lag smoothing, Automatica, 37(2):239–249, 2001.

[59] G. Chen, D. Shen, C. Kwan, J. B. Cruz, Jr., M. Kruger, Game theoretic approach to threat prediction and situation awareness, in: Proceedings of the International Conference on Information Fusion, Paper 245, 2006.

[60] K. Chen, X.-S. Luo, X. Liu, Sensor resource management research based on intelligent agent in naval multi-platform cooperative engagement, in: Proceedings of the International Conference on Machine Learning and Cybernetics, 1:132–136, 2005.

[61] L. Chen, S. Li, IMM tracking of a 3D maneuvering target with passive TDOA system, in: Proceedings of the IEEE International Conference on Neural Networks and Signal Processing, 2:1611–1614, 2003.

[62] X.-Z. Chen, Q.-H. Zheng, X.-H. Guan, C.-G. Lin, J. Sun, Multiple behavior information fusion based quantitative threat evaluation, Computers & Security, 24(3):218–231, 2005.

[63] S. Chin, J. Hill, X. Shen, Opportunistic sensor resource management for extended regions, in: Proceedings of the International Conference on Information Fusion, 2:1138–1145, 2005.

[64] S.-B. Cho, Incorporating soft computing techniques into a probabilistic intrusion detection system, IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, 32(2):154–160, 2002.

[65] J. W. Choi, J. W. Joo, D. L. Cho, Situation/threat assessment fusion system (STAFS), in: Proceedings of the International Conference on Information Fusion, 2:1374–1380, 2002.

[66] C.-Y. Chong, M. Liggins II, Fusion technologies for drug interdiction, in: Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 435–441, 1994.

[67] J. P. Coyne, Bombology, Air Force Magazine, 73(6):64–69, 1990.

[68] N. Cui, L. Hong, J. R. Layne, A comparison of nonlinear filtering approaches with an application to ground target tracking, Signal Processing, 85(8):1469-1492, 2005.

[69] I. W. Dall, Threat assessment without situation assessment, in: Proceedings of International Conference on Information, Decision and Control, pp. 365–370, 1999.

[70] F. Dambreville, Evaluation of a sentry system against noisy optimal intrusions, in: Proceedings of the International Conference on Information Fusion, Session RS07, 2008.

[71] F. Dambreville, J.-P. Le Cadre, Spatio-temporal multi-mode information management for moving target detection, Information Fusion, 5(3):169–178, 2004.

[72] S. Das, R. Grey, P. Gonsalves, Situation assessment via Bayesian belief networks, in: Proceedings of the International Conference on Information Fusion, 1:664–671, 2002.

[73] S. Das, D. Lawless, Truth maintenance system with probabilistic constraints for enhanced level two fusion, in: Proceedings of the International Conference on Information Fusion, 2:993–1000, 2005.

[74] B. V. Dasarathy, Decision Fusion, IEEE Computer Society Press, ISBN: 0818644524, 1994.

[75] B. V. Dasarathy, Sensor fusion potential exploitation - innovative architectures and illustrative applications, Proceedings of the IEEE, 85(1):24–38, 1997.

[76] B. V. Dasarathy, Industrial applications of multi-sensor multi-source information fusion, in: Proceedings of the IEEE International Conference on Industrial Technology, 1:5–11, 2000.

[77] B. V. Dasarathy, Information fusion - what, where, why, when, and how?, Information Fusion, 2(2):75–76, 2001.

[78] B. V. Dasarathy, Information fusion in the context of biomedical applications, Information Fusion, 3(2):89, 2002.

[79] B. V. Dasarathy, Intrusion detection, Information Fusion, 4(4):243–245, 2003.

[80] B. V. Dasarathy, Information fusion in the context of human-machine interfaces, Information Fusion, 6(2):117–118, 2005.

[81] S. G. Dastidar, K. Sambhoos, J. Llinas, C. Bowman, Performance evaluation methods for data-fusion capable tactical platforms, in: Proceedings of the International Conference on Information Fusion, 2:1427–1435, 2005.

[82] F. Daum, Nonlinear filters: beyond the Kalman filter, IEEE Aerospace and Electronic Systems Magazine, 20(8, Part 2):57–69, 2005.

[83] L. J. De Vin, S. F. Andler, A. H. C. Ng, P. R. Moore, J. Pu, B. C.-B. Wong, Information fusion: what can the manufacturing sector learn from the defence industry? in: Proceedings of the International Manufacturing Conference, pp. 363–371, 2005.

[84] L. J. De Vin, A. H. C. Ng, J. Oscarsson, S. F. Andler, Information fusion for simulation based decision support in manufacturing, in: Proceedings of the International Conference on Flexible Automation and Intelligent Manufacturing, pp. 136–144, 2005.

[85] X.-l. Deng, J.-y. Xie, H.-w. Ni, Interacting multiple model algorithm with the unscented particle filter (UPF), Chinese Journal of Aeronautics, 18(4):366–371, 2005.

[86] S. S. Dhillon, K. Chakrabarty, S. S. Iyengar, Sensor placement for grid coverage under imprecise detections, in: Proceedings of the International Conference on Information Fusion, 2:1581–1587, 2002.

[87] A. Doucet, N. de Freitas, N. Gordon (editors), Sequential Monte Carlo Methods in Practice, Springer, ISBN: 0387951466, 2001.

[88] E. Dorion, C. J. Matheus, M. M. Kokar, Towards a formal ontology for military coalitions operations, in: Proceedings of the International Command and Control Research and Technology Symposium, 2005.

[89] H. Driessen, Y. Boers, Efficient particle filter for jump Markov nonlinear systems, IEE Proceedings - Radar, Sonar and Navigation, 152(5):323–326, 2005.

[90] E. Duflos, M. de Vilmorin, P. Vanheeghe, Time allocation of a set of radars in a multitarget environment, in: Proceedings of the International Conference on Information Fusion, Paper 1260, 2007.

[91] A. J. Elbirt, Information warfare: are you at risk?, IEEE Technology and Society Magazine, 22(4):13–19, 2003/2004.

[92] M. R. Endsley, Towards a theory of situation awareness in dynamic systems, Human Factors, 37(l):32–64, 1995.

[93] M. R. Endsley, Theoretical underpinnings of situation awareness: a critical review, in: M. R. Endsley, D. J. Garland (editors), Situation Awareness: Analysis and Measurement, Lawrence Earlbaum Associates, ISBN: 0805821333, pp. 3–32, 2000.

[94] E. B. Entin, The effects of dynamic updating of tactical information on situation awareness and performance in an attack helicopter domain, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 4:3602–3607, 1998.

[95] K. Er, U. R. Acharya, N. Kannathal, C. M. Lim, Data fusion of multimodal cardiovascular signals, in: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4689–4692, 2005.

[96] J. Esteban, A. Starr, R. Willetts, P. Hannah, P. Bryanston-Cross, A review of data fusion models and architectures: towards engineering guidelines, Neural Computing and Applications, 14(4):273–281, 2005.

[97] X. Fan, M. J. Zuo, Fault diagnosis of machines based on D-S evidence theory. Part 1: D-S evidence theory and its improvement, Pattern Recognition Letters, 27(5):366–376, 2006.

[98] X. Fan, M. J. Zuo, Fault diagnosis of machines based on D-S evidence theory. Part 2: application of the improved D-S evidence theory in gearbox fault diagnosis, Pattern Recognition Letters, 27(5):377–385, 2006.

[99] B. Fassinut-Mombot, J.-B. Choquel, A new probabilistic and entropy fusion approach for management of information sources, Information Fusion, 5(1):35–47, 2004.

[100] D. Fava, J. Holsopple, S. J. Yang, B. Argauer, Terrain and behavior modeling for projecting multistage cyber attacks, in: Proceedings of the International Conference on Information Fusion, Paper 1384, 2007.

[101] P. H. Foo, G. W. Ng, Combining IMM method with particle filters for 3D maneuvering target tracking, in: Proceedings of the International Conference on Information Fusion, Paper 1131, 2007.

[102] P. H. Foo, G. W. Ng, K. H. Ng, R. Yang, Application of intent inference for surveillance and conformance monitoring to aid human cognition, in: Proceedings of the International Conference on Information Fusion, Paper 1309, 2007.

[103] G. Fountain, S. Drager, High performance real-time fusion architecture, in: Proceedings of the International Conference on Information Fusion, 2:1478–1485, 2002.

[104] J. Franke, S. M. Brown, B. Bell, H. Mendenhall, Enhancing teamwork through team-level intent inference, in: Proceedings of the International Conference on Artificial Intelligence, 2000.

[105] A. Gad, M. Farooq, Data fusion architecture for maritime surveillance, in: Proceedings of the International Conference on Information Fusion, 1:448–455, 2002.

[106] S. R. Ganta, J. Kasturi, J. Gilbertson, R. Acharya, An online analysis and information fusion platform for heterogeneous biomedical informatics data, in: Proceedings of the IEEE Symposium on Computer-Based Medical Systems, pp. 153–158, 2005.

[107] B. J. Garrick, J. E. Hall, M. Kilger, J. C. McDonald, T. O'Toole, P. S. Probst, E. R. Parker, R. Rosenthal, A. W. Trivelpiece, L. A. Van Arsdale, E. L. Zebroski, Confronting the risks of terrorism: making the right decisions, Reliability Engineering and System Safety, 86(2):129–176, 2004.

[108] R. Ghanadan, P. Tufano, J. Hsu, J. Gu, C. Connelly, Flexible access secure transfer (FAST) tactical communications waveform for airborne networking, in: Proceedings of the IEEE Military Communications Conference, pp. 1–7, 2005.

[109] R. Glinton, J. Giampapa, K. Sycara, A Markov random field model of context for high-level information fusion, in: Proceedings of the International Conference on Information Fusion, Paper 375, 2006.

[110] K. F. Goebel, W. Yan, Fusing binary and continuous output of multiple classifiers, in: Proceedings of the International Conference on Information Fusion, 1:380–387, 2002.

[111] G. H. Golub, C. F. Van Loan, Matrix Computations (3rd edition), Johns Hopkins University Press, ISBN: 080185413X, 1996.

[112] P. G. Gonsalves, J. E. Burge, K. A. Harper, Architecture for genetic algorithm-based threat assessment, in: Proceedings of the International Conference on Information Fusion, 2:965–971, 2003.

[113] P. G. Gonsalves, R. Cunningham, N. Ton, D. Okon, Intelligent threat assessment processor (ITAP) using genetic algorithms and fuzzy logic, in: Proceedings of the International Conference on Information Fusion, 2:ThB1-18–ThB1-24, 2000.

[114] P. G. Gonsalves, G. J. Rinkus, Intelligent fusion and asset management processor, in: Proceedings of the IEEE Information Technology Conference, pp. 15–18, 1998.

[115] A. A. H. E. Goossens, G. J. M. Koeners, E. Theunissen, Development and evaluation of level 3 situation awareness support functions for a UAV operator station, in: Proceedings of the Digital Avionics Systems Conference, 2:12.D.4-1–12.D.4-8, 2004.

[116] N. J. Gordon, D. J. Salmond, A. F. M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proceedings F, Radar Signal Processing, 140(2):107–113, 1993.

[117] V. Gorodetsky, O. Karsaev, V. Samoilov, On-line update of situation assessment: a generic approach, International Journal of Knowledge-Based and Intelligent Engineering Systems, 9(4):351–365, 2005.

[118] J. H. Graham, Y. Yu, Computer system security threat evaluation based upon artificial immunity model and fuzzy logic, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2:1297–1302, 2005.

[119] Y.-f. Guo, W. Wei, A.-k. Xue, D.-c. Mao, A filter algorithm for multi-measurement nonlinear system with parameter perturbation, Journal of Zhejiang University - Science A, 7(10):1717–1722, 2006.

[120] F. Gustafsson, F. Gunnarsson, Positioning using time-difference of arrival measurements, in: Proceedings of International Conference on Acoustic, Speech, and Signal Processing, 6:VI-553–VI-556, 2003.

[121] D. L. Hall, J. Llinas, An introduction to multisensor data fusion, Proceedings of the IEEE, 85(1):6–23, 1997.

[122] D. L. Hall, J. Llinas (editors), Handbook of Multisensor Data Fusion, CRC Press, ISBN: 0849323797, 2001.

[123] D. L. Hall, S. A. H. McMullen, Mathematical Techniques in Multisensor Data Fusion (2nd edition), Artech House, ISBN: 1580533353, 2004.

[124] D. L. Hall, M. D. McNeese, E. Rotthoff, T. Shaw, J. Z. Wang, Improving the fusion process using semantic level whole-brain analysis, in: Proceedings of the Military Sensing Series National Symposium on Sensor and Data Fusion, 2005.

[125] M. J. M. Hall, S. A. Hall, T. Tate, Removing the HCI bottleneck: how the human computer interface (HCI) affects the performance of data fusion systems, in: [122].

[126] E. M. Hamed, T. S. Sobh, A flexible fuzzy threat evaluation computer system, in: Proceedings of the International Conference on Electrical, Electronic and Computer Engineering, pp. 23–27, 2004.

[127] J. M. Hammersley, K. W. Morton, Poor man's Monte Carlo, Journal of the Royal Statistical Society Series B (Methodological), 16(1):23–38, 1954.

[128] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, W. V. Seelen, An image processing system for driver assistance, Image and Vision Computing, 18(5):367–376, 2000.

[129] A. J. Haug, A Tutorial on Bayesian Estimation and Tracking Techniques Applicable to Nonlinear and Non-Gaussian Processes, Technical Report MTR 05W0000004, The MITRE Corporation, USA, 2005.

[130] U. Heiden, K. Segl, S. Roessner, H. Kaufmann, Ecological evaluation of urban biotope types using airborne hyperspectral HyMap data, in: Proceedings of the Joint Workshop on Remote Sensing and Data Fusion over Urban Areas, pp. 18–22, 2003.

[131] A. O. Hero III, D. A. Castañón, D. Cochran, K. Kastella (editors), Foundations and Applications of Sensor Management, Springer, ISBN: 9780387278926, 2008.

[132] A. O. Hero III, C. M. Kreucher, Network sensor management for tracking and localization, in: Proceedings of the International Conference on Information Fusion, Paper 1459, 2007.

[133] J. P. Hill, K. C. Chang, Improved representations of sensor exploitation for automatic sensor management, in: Proceedings of the International Conference of Information Fusion, 1:688–694, 2003.

[134] J. P. Hill, K. C. Chang, Sensor resource management with hierarchical target valuation models, in: Proceedings of the International Conference on Information Fusion, 2:1187–1194, 2004.

[135] J. P. Hill, K. C. Chang, Sensor resource management with level 2 fusion using Markov chain models, in: Proceedings of the International Conference on Information Fusion, 2:1444–1450, 2005.

[136] F. Hinchion, P. Mulgaonkar, D. Wilkins, S. Galuga, Peer to peer information management for tactical situation awareness systems, in: Proceedings of the IEEE Military Communications Conference, 1:179–185, 2003.

[137] M. L. Hinman, Some computational approaches for situation assessment and impact assessment, in: Proceedings of the International Conference on Information Fusion, 1:687–693, 2002.

[138] M. J. Hirsch, R. Nagi, D. Sudit, A stochastic optimization framework for resource management and course of action analysis, in: Proceedings of the International Conference on Information Fusion, Session SS04, 2008.

[139] L. Hong, M. Bakich, J. R. Layne, Efficient multirate interacting multiple model particle filter (MRIMM-PF) for target tracking, in: Proceedings of SPIE, 6229:62290S, 2006.

[140] L. Hong, N. Cui, M. Bakich, J.R. Layne, Multirate interacting multiple model particle filter for terrain-based ground target tracking, IEE Proceedings - Control Theory Applications, 153(6):721–731, 2006.

[141] P. Hörling (editor), C. Mårtenson, J. Schubert, R. Suzić, P. Svenson, A Survey of Some Information Fusion Research Fields Applicable in Operations Other than War, User Report FOI-R–2000–SE, the Swedish Defence Research Agency, Sweden, 2006.

[142] C. Y. Huang, Real-time automated decision-making in advanced airborne early warning system, in: Proceedings of the IEEE National Aerospace and Electronics Conference, 1:434–439, 1990.

[143] L. Hubert-Moy, S. Corgne, G. Mercier, B. Solaiman, Land use and land cover change prediction with the theory of evidence: a case study in an intensive agricultural region of France, in: Proceedings of the International Conference on Information Fusion, 1:114–121, 2002.

[144] L. D. Hudson, B. S. Ware, K. B. Laskey, S. M. Mahoney, An Application of Bayesian Networks to Antiterrorism Risk Management for Military Planners, Technical Report C4I-05-01, Center of Excellence in Command, Control, Communications, Computing and Intelligence (C4I Center), George Mason University, USA, 2001.

[145] C. Hue, J.-P. Le Cadre, P. Pérez, Sequential Monte Carlo methods for multiple target tracking and data fusion, IEEE Transactions on Signal Processing, 50(2):309–325, 2002.

[146] J. C. Hull, Options, Futures, and Other Derivatives (6th edition), Prentice Hall, ISBN: 0131977059, 2006.

[147] I. Hwang, H. Balakrishnan, K. Roy, C. Tomlin, Multiple-target tracking and identity management algorithm with application to aircraft tracking, Journal of Guidance, Control and Dynamics, 30(3):641–653, 2007.

[148] R. Iannella, Modeling and integration of severe weather advisories for situational awareness, in: Proceedings of the International Conference on Information Fusion, Paper 22, 2006.

[149] J. E. Introne, I. Levit, S. Harrison, S. Das, A data fusion approach to biosurveillance, in: Proceedings of the International Conference on Information Fusion, 2:1359–1366, 2005.

[150] K. Ito, K. Xiong, Gaussian filters for nonlinear filtering problems, IEEE Transactions on Automatic Control, 45(5):910–927, 2000.

[151] T. Jan, Neural network based threat assessment for automated visual surveillance, in: Proceedings of the IEEE International Joint Conference on Neural Networks, 2:1309–1312, 2004.

[152] J.-S. R. Jang, C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Prentice Hall, ISBN: 0132610663, 1997.

[153] P. Jarvis, T. F. Lunt, K. L. Myers, Identifying terrorist activity with AI plan recognition technology, in: Proceedings of the National Conference on Artificial Intelligence, Conference on Innovative Applications of Artificial Intelligence, pp. 858–863, 2004.

[154] F. Johansson, G. Falkman, A Bayesian network approach to threat evaluation with application to an air defense scenario, in: Proceedings of the International Conference on Information Fusion, Session RS21, 2008.

[155] L. R. M. Johansson, C. Mårtenson, R. Suzić, P. Svenson, Stochastic Dynamic Programming for Resource Allocation, Technical Report FOI-R–1666–SE, the Swedish Defence Research Agency, Sweden, 2005.

[156] L. R. M. Johansson, N. Xiong, Perception management: an emerging concept for information fusion, Information Fusion, 4(3):231–234, 2003.

[157] W. T. Johnson, I. W. Dall, From kinematics to symbolics for situation and threat assessment, in: Proceedings of International Conference on Information, Decision and Control, pp. 497–502, 1999.

[158] S. J. Julier, J. K. Uhlmann, Data fusion in nonlinear systems, in: [122].

[159] I. Kadar, Knowledge representation issues in perceptual reasoning managed situation assessment, in: Proceedings of the International Conference on Information Fusion, 1:xiii–xv, 2005.

[160] N. Kannathal, U. R. Acharya, E. Y. K. Ng, S. M. Krishnan, C. M. Lim, S. Laxminarayan, Cardiac health diagnosis using data fusion of cardiovascular and haemodynamic signals, Computer Methods and Programs in Biomedicine, 82(2):87–96, 2006.

[161] A. Karlsson, Dependable and Generic High-level Information Fusion - Methods and Algorithms for Uncertainty Management, Technical Report HS-IKI-TR-07-003, School of Humanities and Informatics, University of Skövde, Sweden, 2007.

[162] A. Khalil, M. K. Gill, M. McKee, New applications for information fusion and soil moisture forecasting, in: Proceedings of the International Conference on Information Fusion, 2:1622–1628, 2005.

[163] L. A. Klein, Sensor and Data Fusion: A Tool for Information Assessment and Decision Making, SPIE Press, ISBN: 0819454354, 2004.

[164] M. M. Kokar, Situation awareness: issues and challenges, in: Proceedings of the International Conference on Information Fusion, 1:533–534, 2004.

[165] M. M. Kokar, Situation assessment: procedural vs. logical, in: Proceedings of the International Conference on Information Fusion, 1:xx–xxiii, 2005.

[166] M. M. Kokar, M. D. Bedworth, C. B. Frankel, A reference model for data fusion systems, in: Proceedings of SPIE, 4051:191–202, 2000.

[167] M. M. Kokar, C. J. Matheus, K. Baclawski, Ontology-based situation awareness, Information Fusion, 10(1):83–98, 2009.

[168] M. M. Kokar, C. J. Matheus, K. Baclawski, J. A. Letkowski, M. Hinman, J. Salerno, Use cases for ontologies in information fusion, in: Proceedings of the International Conference on Information Fusion, 1:415–422, 2004.

[169] M. M. Kokar, C. J. Matheus, J. A. Letkowski, K. Baclawski, P. Kogut, Association in level 2 fusion, in: Proceedings of SPIE, 5434:228–237, 2004.

[170] M. M. Kokar, G. W. Ng, High-level information fusion and situation awareness, Information Fusion, 10(1):2–5, 2009.

[171] M. M. Kokar, J. Wang, An example of using ontologies and symbolic information in automatic target recognition, in: Proceedings of SPIE, 4731:40–50, 2002.

[172] M. M. Kokar, J. Wang, Using ontologies for recognition: an example, in: Proceedings of the International Conference on Information Fusion, 2:1324–1343, 2002.

[173] W. Komorniczak, J. Pietrasinski, B. Solaiman, The data fusion approach to the priority assignment in the multifunction radar, in: Proceedings of the International Conference on Microwaves, Radar and Wireless Communications, 2:647–650, 2002.

[174] J. H. Kotecha, P. M. Djurić, Gaussian particle filtering, IEEE Transactions on Signal Processing, 51(10):2592–2601, 2003.

[175] J. H. Kotecha, P. M. Djurić, Gaussian sum particle filtering, IEEE Transactions on Signal Processing, 51(10):2602–2612, 2003.

[176] A. Kott (editor), Information Warfare and Organizational Decision-making, Artech House, ISBN: 1596930799, 2007.

[177] A. Kott, W. M. McEneaney (editors), Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind, Taylor & Francis, ISBN: 1584885882, 2006.

[178] C. Kreucher, D. Blatt, A. Hero, K. Kastella, Adaptive multi-modality sensor scheduling for detection and tracking of smart targets, Digital Signal Processing, 16(5):546–567, 2006.

[179] C. Kreucher, K. Kastella, A. O. Hero III, Sensor management using an active sensing approach, Signal Processing, 85(3):607–624, 2005.

[180] C. M. Kreucher, M. Morelande, K. Kastella, A. O. Hero III, Joint multi-target particle filtering, in: [131].

[181] J. Krozel, D. Andrisani II, Intent inference with strategic path prediction, Journal of Guidance, Control, and Dynamics, 29(2):225–236, 2006.

[182] J. Krozel, T. Mueller, D. Schleicher, Intent Inference, Confidence Assessment, and Hazard Prioritization Status Report, Technical Report TR 99175.30-02, Seagull Technology, Inc., USA, 2000.

[183] G. G. Kuperman, R. D. Whitaker, S. M. Brown, "Cyber Warrior": information superiority through advanced multi-sensory command and control technologies, in: Proceedings of the IEEE National Aerospace and Electronics Conference, pp. 263–271, 2000.

[184] D. A. Lambert, Grand challenges of information fusion, in: Proceedings of the International Conference on Information Fusion, 1:213–220, 2003.

[185] D. A. Lambert, A unification of sensor and higher-level fusion, in: Proceedings of the International Conference on Information Fusion, Paper 79, 2006.

[186] D. A. Lambert, STDF model based maritime situation assessments, in: Proceedings of the International Conference on Information Fusion, Paper 1269, 2007.

[187] D. A. Lambert, A blueprint for higher-level fusion systems, Information Fusion, 10(1):6–24, 2009.

[188] J. Landis, L. Bai, J. Salerno, M. Hinman, D. Boulware, Mobile agent-based distributed fusion (MADFUSION) system, in: Proceedings of the International Conference on Information Fusion, 2:1199–1203, 2005.

[189] K. B. Laskey, P. C. G. Costa, T. Janssen, Probabilistic ontologies for knowledge fusion, in: Proceedings of the International Conference on Information Fusion, Session RS08, 2008.

[190] K. D. Lee, A. E. Gelfand, E. Wiesenfeld, B. Stepnitz, Introduction of the Hybrid Inference Tool (HIT), in: Proceedings of the International Conference on Information Fusion, Paper 1365, 2007.

[191] X.-R. Li, V. P. Jilkov, A survey of maneuvering target tracking-Part II: ballistic target models, in: Proceedings of SPIE, 4473:559–581, 2001.

[192] X.-R. Li, V. P. Jilkov, A survey of maneuvering target tracking-Part III: measurement models, in: Proceedings of SPIE, 4473:423–446, 2001.

[193] X.-R. Li, V. P. Jilkov, A survey of maneuvering target tracking-Part IV: decision-based methods, in: Proceedings of SPIE, 4728:511–534, 2002.

[194] X.-R. Li, V. P. Jilkov, Survey of maneuvering target tracking. Part I: dynamic models, IEEE Transactions on Aerospace and Electronic Systems, 39(4):1333–1364, 2003.

[195] X.-R. Li, V. P. Jilkov, A survey of maneuvering target tracking: approximation techniques for nonlinear filtering, in: Proceedings of SPIE, 5428:537–550, 2004.

[196] X.-R. Li, V. P. Jilkov, Survey of maneuvering target tracking. Part V: multiple-model methods, IEEE Transactions on Aerospace and Electronic Systems, 41(4):1255–1321, 2005.

[197] M. E. Liggins II, A. Bramson, Off-board augmented fusion for improved target detection and track, in: Proceedings of the Asilomar Conference on Signals, Systems, and Computers, pp. 295–299, 1993.

[198] M. E. Liggins II, C.-Y. Chong, Distributed multi-platform fusion for enhanced radar management, in: Proceedings of the IEEE National Radar Conference, pp. 115–119, 1997.

[199] E. G. Little, G. L. Rogova, Ontology meta-model for building a situational picture of catastrophic events, in: Proceedings of the International Conference on Information Fusion, 1:796–803, 2005.

[200] E. G. Little, G. L. Rogova, An ontological analysis of threat and vulnerability, in: Proceedings of the International Conference on Information Fusion, Paper 302, 2006.

[201] E. G. Little, G. L. Rogova, Designing ontologies for higher level fusion, Information Fusion, 10(1):70–82, 2009.

[202] E. Little, K. Sambhoos, J. Llinas, Enhancing graph matching techniques with ontologies, in: Proceedings of the International Conference on Information Fusion, Session SS14, 2008.

[203] J. Llinas, Developing a cost-effective approach to information fusion research and development, in: Proceedings of International Conference on Information, Decision and Control, pp. 511, 1999.

[204] J. Llinas, Information fusion for natural and man-made disasters, in: Proceedings of the International Conference on Information Fusion, 1:570–576, 2002.

[205] J. Llinas, C. Bowman, G. L. Rogova, A. Steinberg, E. Waltz, F. White, Revisiting the JDL data fusion model II, in: Proceedings of the International Conference on Information Fusion, 2:1218–1230, 2004.

[206] J. Llinas, M. D. Moskal, T. McMahon, Information fusion for nuclear, chemical, biological & radiological (NCBR) battle management support/disaster response management support, Center for MultiSource Information Fusion, School of Engineering and Applied Sciences, University of Buffalo, USA, 2002.

[207] J. Llinas, D. Talreja, C. Bowman, Design of a performance evaluation methodology for data fusion-based multiple target tracking systems II: analysis methodology, in: Proceedings of the Military Sensing Series National Symposium on Sensor and Data Fusion, 2004.

[208] C. G. Looney, L. R. Liang, Cognitive situation and threat assessments of ground battlespaces, Information Fusion, 4(4):297–308, 2003.

[209] G. B. Magklaras, S. M. Furnell, Insider threat prediction tool: evaluating the probability of IT misuse, Computers & Security, 21(1):62–73, 2002.

[210] R. Mahler, Unified sensor management using CPHD filters, in: Proceedings of the International Conference on Information Fusion, Paper 1172, 2007.

[211] R. P. S. Mahler, Statistical Multisource-Multitarget Information Fusion, Artech House, ISBN: 1596930926, 2007.

[212] B. Majidi, B. Moshiri, Industrial assessment of horticultural products' quality using image data fusion, in: Proceedings of the International Conference on Information Fusion, 2:868–873, 2003.

[213] M. Mandiak, Y. Kim, P. Shah, T. Kesavadas, Development of an integrated GUI framework for post-disaster data fusion visualization, in: Proceedings of the International Conference on Information Fusion, 2:1131–1137, 2005.

[214] C. Mårtenson, P. Svenson, Evaluating sensor allocations using equivalence classes of multitarget paths, in: Proceedings of the International Conference on Information Fusion, 1:574–581, 2005.

[215] M. A. Masnadi-Shirazi, S. A. Banani, On maneuvering target tracking with online observed colored glint noise parameter estimation, in: International Journal of Applied Mathematics and Computer Sciences, 4(2):111–117, 2008.

[216] C. J. Matheus, Using ontology-based rules for situation awareness and information fusion, in: Proceedings of the W3C Workshop on Rule Languages for Interoperability, 2005.

[217] C. J. Matheus, K. Baclawski, M. M. Kokar, Derivation of ontological relations using formal methods in a situation awareness scenario, in: Proceedings of SPIE, 5099:298–309, 2003.

[218] C. J. Matheus, M. M. Kokar, K. Baclawski, A core ontology for situation awareness, in: Proceedings of the International Conference on Information Fusion, 1:545–552, 2003.

[219] C. J. Matheus, M. M. Kokar, K. Baclawski, J. J. Letkowski, C. Call, M. Hinman, J. Salerno, D. Boulware, SAWA: an assistant for higher-level fusion and situation awareness, in: Proceedings of SPIE, 5813:75–85, 2005.

[220] C. J. Matheus, M. M. Kokar, K. Baclawski, J. J. Letkowski, C. Call, M. Hinman, J. Salerno, D. Boulware, Lessons learned from developing SAWA: a situation awareness assistant, in: Proceedings of the International Conference on Information Fusion, 2:969–976, 2005.

[221] J. Mattioli, N. Museux, M. Hemaissia, C. Laudy, A crisis response situation model, in: Proceedings of the International Conference on Information Fusion, Paper 1217, 2007.

[222] E. Mazor, A. Averbuch, Y. Bar-Shalom, J. Dayan, Interacting multiple model methods in target tracking: a survey, IEEE Transactions on Aerospace Electronic Systems, 34(1):103–123, 1998.

[223] D. McMichael, G. Jarrad, S. Williams, M. Kennett, Force deployment analysis with generalized grammar, Information Fusion, 10(1):51–69, 2009.

[224] J. Molina, J. García, F. J. Jiménez, J. R. Casar, Surveillance multisensor management with fuzzy evaluation of sensor task priorities, Engineering Applications of Artificial Intelligence, 15(6):511–527, 2002.

[225] R. Moskovitch, N. Nissim, R. Englert, Y. Elovici, Active learning to improve the detection of unknown computer worms activity, in: Proceedings of the International Conference on Information Fusion, Session RS07, 2008.

[226] C. Muller, M. Rombaut, M. Janier, Dempster Shafer approach for high level data fusion applied to the assessment of myocardial viability, Lecture Notes in Computer Science, 2230:104–112, 2001.

[227] D. Mušicki, Target existence based resource allocation, in: Proceedings of the International Conference on Information Fusion, Paper 1258, 2007.

[228] R. Nagi, M. Sudit, J. Llinas, An approach for level 2/3 fusion technology development in urban/asymmetric scenarios, in: Proceedings of the International Conference on Information Fusion, Paper 236, 2006.

[229] M. Negnevitsky, Artificial Intelligence: a Guide to Intelligent Systems (2nd edition), Addison-Wesley, ISBN: 0321204662, 2005.

[230] G. W. Ng, Intelligent Systems - Fusion, Tracking and Control, Research Studies Press Ltd. and Institute of Physics Publishing, ISBN: 086380277X, 2003.

[231] G. W. Ng, K. H. Ng, Sensor management - what, why and how, Information Fusion, 1(2):67–75, 2000.

[232] G. W. Ng, K. H. Ng, K. H. Tan, C. H. K. Goh, The ultimate challenge of commander's decision aids: the cognition based dynamic reasoning machine, in: Proceedings of the Army Science Conference, Paper BO-05, 2006.

[233] G. W. Ng, K. H. Ng, K. H. Tan, C. H. K. Goh, Novel methods for fusing Bayesian network knowledge fragments in D'Brain, in: Proceedings of the International Conference on Information Fusion, Paper 1305, 2007.

[234] G. W. Ng, K. H. Ng, R. Yang, P. H. Foo, Intent inference for attack aircraft through fusion, in: Proceedings of SPIE, 6242:624206, 2006.

[235] G. W. Ng, C. H. Tan, T. P. Ng, S. Y. Siow, Assessment of data fusion systems, in: Proceedings of the International Conference on Information Fusion, Paper 63, 2006.

[236] X. T. Nguyen, Threat assessment in tactical airborne environments, in: Proceedings of the International Conference on Information Fusion, 2:1300–1307, 2002.

[237] D. M. Nicol, Modeling and simulation in security evaluation, IEEE Security & Privacy, 3(5):71–74, 2005.

[238] M. Nilsson, Human Decision Making and Information Fusion: Extending the Concept of Decision Support, Technical Report HS-IKI-TR-07-002, School of Humanities and Informatics, University of Skövde, Sweden, 2007.

[239] M. Nilsson, T. Ziemke, Rethinking level 5: distributed cognition and information fusion, in: Proceedings of the International Conference on Information Fusion, Paper 98, 2006.

[240] M. Niranjan, Sequential tracking in pricing financial options using model based and neural network approaches, Advances in Neural Information Processing Systems, 9:960–966, 1997.

[241] C. Nowak, On ontologies for high-level information fusion, in: Proceedings of the International Conference on Information Fusion, 1:657–664, 2003.

[242] C. Nowak, D. A. Lambert, The semantic challenge for situation assessments, in: Proceedings of the International Conference on Information Fusion, 1:804–811, 2005.

[243] S. Oh, S. Sastry, An efficient algorithm for tracking multiple maneuvering targets, in: Proceedings of the IEEE Conference on Decision and Control, and the European Control Conference, pp. 4010–4015, 2005.

[244] N. Okello, G. Thorns, Threat assessment using Bayesian networks, in: Proceedings of the International Conference on Information Fusion, 2:1102–1109, 2003.

[245] B. Olsson, P. Nilsson, B. Gawronska, A. Persson, T. Ziemke, S. F. Andler, An information fusion approach to controlling complexity in bioinformatics research, in: Proceedings of the IEEE Computational Systems Bioinformatics Conference Workshops, pp. 299–304, 2005.

[246] J. K. Ong, J. Ibañez-Guzmán, Perception management for the guidance of unmanned vehicles, in: Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, 2:914–919, 2004.

[247] M. Oxenham, Enhancing situation awareness for air defence via automated threat analysis, in: Proceedings of the International Conference on Information Fusion, 2:1086–1093, 2003.

[248] M. Oxenham, S. Challa, M. Morelande, Decentralised fusion of disparate identity estimates for shared situation awareness, in: Proceedings of the International Conference on Information Fusion, 1:167–174, 2004.

[249] M. Oxenham, S. Challa, M. Morelande, Fusion of disparate identity estimates for shared situation awareness in a network-centric environment, Information Fusion, 7(4):395–417, 2006.

[250] M. G. Oxenham, D. J. Kewley, M. J. Nelson, Performance assessment of data fusion systems, in: Proceedings of the Australian Data Fusion Symposium, pp. 36–41, 1996.

[251] S. Paradis, A. Benaskeur, M. Oxenham, P. Cutler, Threat evaluation and weapons allocation in network-centric warfare, in: Proceedings of the International Conference on Information Fusion, 2:1078–1085, 2005.

[252] O. Parsons, G. A. Carpenter, ARTMAP neural networks for information fusion and data mining: map production and target recognition methodologies, Neural Networks, 16(7):1075–1089, 2003.

[253] G. Petterson, L. Axelsson, T. Jensen, M. Karlsson, A. Malmberg, Multi-source integration and temporal situation assessment in air combat, in: Proceedings of International Conference on Information, Decision and Control, pp. 371–375, 1999.

[254] A. Pfeffer, S. Das, D. Lawless, B. Ng, Factored reasoning for monitoring dynamic team and goal formation, Information Fusion, 10(1):99–106, 2009.

[255] T. Pham, G. H. Cirincione, D. Verma, G. Pearson, Intelligence, surveillance, and reconnaissance fusion for coalition operations, in: Proceedings of the International Conference on Information Fusion, Session SS19, 2008.

[256] M. K. Pitt, N. Shephard, Filtering via simulation: auxiliary particle filters, Journal of the American Statistical Association, 94(446):590–599, 1999.

[257] M. T. Plehn, Control warfare: inside the OODA loop, Master's thesis, School of Advanced Airpower Studies, Air University, Maxwell Air Force Base, USA, 2000.

[258] A. Polychronopoulos, A. Amditis, U. Scheunert, T. Tatschke, Revisiting JDL model for automotive safety applications: the PF2 functional model, in: Proceedings of the International Conference on Information Fusion, Paper 259, 2006.

[259] R. L. Popp, A. W. Bailey, J. N. Tsitsiklis, Dynamic airborne sensor resource management for ground moving target tracking and classification, in: Proceedings of the IEEE Aerospace Conference, 3:405–415, 2000.

[260] G. M. Powell, Tactical situation assessment challenges and implications for computational support, in: Proceedings of the International Conference on Information Fusion, 1:xxiv–xxvi, 2005.

[261] S. Puranik, J. K. Tugnait, Tracking of multiple maneuvering targets using multiscan JPDA and IMM filtering, IEEE Transactions on Aerospace and Electronic Systems, 43(1):23–35, 2007.

[262] R. G. Qiu, A data fusion framework for an integrated plant-wide information system, in: Proceedings of the International Conference on Information Fusion, 1:101–107, 2002.

[263] C. Qu, Y. He, A method of threat assessment using multiple attribute decision making, in: Proceedings of the International Conference on Signal Processing, 2:1091–1095, 2002.

[264] Z. Qureshi, P. Urlings, Situation awareness and automation: issues and design approaches, in: Proceedings of International Conference on Information, Decision and Control, pp. 605–610, 1999.

[265] K. V. Ramachandra, Kalman Filtering Techniques for Radar Tracking, Marcel Dekker, ISBN: 0824793226, 1999.

[266] U. Ramdaras, F. Absil, Sensor selection: the modified Riccati equation approach compared with other selection schemes, in: Proceedings of the International Conference on Information Fusion, Paper 1184, 2007.

[267] S. Rawat, J. Llinas, C. Bowman, Design of a performance evaluation methodology for data fusion-based multiple target tracking systems, in: Proceedings of SPIE, 5099:139–151, 2003.

[268] T. G. Reynolds, R. J. Hansman, Analysis of aircraft separation minima using a surveillance state vector approach, in: Proceedings of the FAA/Eurocontrol International Air Traffic Management R&D Seminar, 2000.

[269] T. G. Reynolds, R. J. Hansman, Conformance monitoring approaches in current and future air traffic control environments, in: Proceedings of the IEEE/AIAA Digital Avionics Systems Conference, 2:7.C.1-1–7.C.1-12, 2002.

[270] T. G. Reynolds, R. J. Hansman, Analyzing conformance monitoring in air traffic control using fault detection approaches & operational data, in: Proceedings of the AIAA Guidance, Navigation, and Control Conference, Paper AIAA-2003-5574, 2003.

[271] T. G. Reynolds, R. J. Hansman, Investigating conformance monitoring issues in air traffic control using fault detection techniques, Journal of Aircraft, 42(5):1307–1317, 2005.

[272] T. G. Reynolds, R. J. Hansman, H. Li, Investigating fundamental issues in lateral conformance monitoring using a fault detection approach, in: Proceedings of the Eurocontrol/FAA Air Traffic Management R&D Seminar, 2003.

[273] T. G. Reynolds, J. M. Histon, H. J. Davison, R. J. Hansman, Structure, intent & conformance monitoring in ATC, in: Proceedings of the Workshop on Advanced Technologies and Their Impact on Air Traffic Management in the 21st Century, 2002.

[274] B. J. Rhodes, Taxonomic knowledge structure discovery from imagery-based data using the neural associative incremental learning (NAIL) algorithm, Information Fusion, 8(3):295–315, 2007.

[275] J. Rickard, Level 2/3 fusion in conceptual spaces, in: Proceedings of the International Conference on Information Fusion, Paper 16, 2006.

[276] B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking and Applications, Artech House, ISBN: 158053631X, 2004.

[277] G. L. Rogova, P. D. Scott, C. Lollett, Higher level fusion for post-disaster casualty mitigation operations, in: Proceedings of the International Conference on Information Fusion, 2:938–945, 2005.

[278] M. N. Rosenbluth, A. W. Rosenbluth, Monte Carlo calculation of the average extension of molecular chains, The Journal of Chemical Physics, 23(2):356–359, 1955.

[279] R. Rousseau, R. Breton, The M-OODA: A model incorporating control functions and teamwork in the OODA loop, in: Proceedings of the Command and Control Research and Technology Symposium, 2004.

[280] S. Roussel, V. Bellon-Maurel, J.-M. Roger, P. Grenier, Fusion of aroma, FT-IR and UV sensor data based on the Bayesian inference. Application to the discrimination of white grape varieties, Chemometrics and Intelligent Laboratory Systems, 65(2):209–219, 2003.

[281] J. N. Roux, J. H. van Vuuren, Threat evaluation and weapon assignment decision support: a review of the state of the art, ORiON, 23(2):151–187, 2007.

[282] V. Royer, J.-F. Challine, A platform for interoperable fusion models, in: Proceedings of the International Conference on Information Fusion, 2:ThB1-3–ThB1-10, 2000.

[283] Y. Rui, Y. Chen, Better proposal distributions: object tracking using unscented particle filter, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. II-786–II-793, 2001.

[284] V. Sakkalis, M. Zervakis, S. Micheloyannis, Biopattern initiative: towards the development and integration of next-generation information fusion approaches, in: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1:399–402, 2004.

[285] J. Salerno, Information fusion: a high-level architecture overview, in: Proceedings of the International Conference on Information Fusion, 1:680–686, 2002.

[286] J. Salerno, M. Hinman, D. Boulware, P. Bello, Information fusion for situational awareness, in: Proceedings of the International Conference on Information Fusion, 1:507–513, 2003.

[287] P. Salmon, N. Stanton, G. Walker, D. Green, Situation awareness measurement: a review of applicability for C4i environments, Applied Ergonomics, 37(2):225–238, 2006.

[288] D. D. Salvucci, Inferring driver intent: a case study in lane-change detection, in: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, pp. 2228–2231, 2004.

[289] K. Sambhoos, M. Temel, F. Pan, M. Sudit, Level 4 fusion for biosensor location and setting, in: Proceedings of the International Conference on Information Fusion, Session SS04, 2008.

[290] E. Santos Jr., A cognitive architecture for adversary intent inferencing: knowledge structure and computation, in: Proceedings of SPIE, 5091:182–193, 2003.

[291] E. Shahbazian, D. E. Blodgett, P. Labbé, The extended OODA model for data fusion systems, in: Proceedings of the International Conference on Information Fusion, pp. FrB1-19–FrB1-25, 2001.

[292] T. Schön, F. Gustafsson, P. Nordlund, Marginalized particle filters for mixed linear/nonlinear state-space models, IEEE Transactions on Signal Processing, 53(7):2279–2289, 2005.

[293] D. Shen, G. Chen, L. Haynes, E. Blasch, Strategies comparison for game theoretic cyber situational awareness and impact assessment, in: Proceedings of the International Conference on Information Fusion, Paper 1317, 2007.

[294] Y. Shu, F. Kazuo, Team situation awareness as a basis of team intention inference: definition and framework, in: Proceedings of the SICE Annual Conference, 3:1641–1646, 2002.

[295] C. Simonin, J.-P. Le Cadre, F. Dambreville, The cross-entropy method for solving a variety of hierarchical search problems, in: Proceedings of the International Conference on Information Fusion, Paper 1483, 2007.

[296] P. R. Smart, N. R. Shadbolt, L. A. Carr, M. C. Schraefel, Knowledge-based information fusion for improved situational awareness, in: Proceedings of the International Conference on Information Fusion, 2:1017–1024, 2005.

[297] A. Smirnov, M. Pashkin, N. Shilov, T. Levashova, A. Kashevnik, Context-aware operational decision support, in: Proceedings of the International Conference on Information Fusion, Paper 1464, 2007.

[298] L. Snidaro, M. Belluz, G. L. Foresti, Domain knowledge for surveillance applications, in: Proceedings of the International Conference on Information Fusion, Paper 1367, 2007.

[299] F. Stein, J. Garska, P. L. McIndoo, Network-centric warfare: impact on army operations, in: Proceedings of the IEEE/AFCEA Conference on Information Systems for Enhanced Public Safety and Security, pp. 288–295, 2000.

[300] A. N. Steinberg, Unification across JDL data fusion levels 1 and 2, in: Proceedings of the International Conference on Information Fusion, 1:530–532, 2004.

[301] A. N. Steinberg, Threat assessment technology development, in: Proceedings of the International and Interdisciplinary Conference on Modeling and Using Context, pp. 490–500, 2005.

[302] A. N. Steinberg, An approach to threat assessment, in: Proceedings of the International Conference on Information Fusion, 2:1256–1263, 2005.

[303] A. N. Steinberg, Predictive modeling of interacting agents, in: Proceedings of the International Conference on Information Fusion, Paper 1136, 2007.

[304] A. N. Steinberg, C. L. Bowman, F. E. White, Revisions to the JDL data fusion model, in: Proceedings of SPIE, 3719:430–441, 1999.

[305] L. D. Stone, C. A. Barlow, T. L. Corwin, Bayesian Multiple Target Tracking, Artech House, ISBN: 1580530249, 1999.

[306] A. Stotz, M. Sudit, INformation Fusion Engine for Real-time Decision-making (INFERD): a perceptual system for cyber attack tracking, in: Proceedings of the International Conference on Information Fusion, Paper 1359, 2007.

[307] S. Stubberud, P. J. Shea, D. Klamer, Metrics for level 2 fusion association, in: Proceedings of the International Conference on Information Fusion, 2:1180–1186, 2003.

[308] S. C. Stubberud, K. A. Kramer, Level 2 fusion: situational assessment composition fusion with uncertain classification, in: Proceedings of the International Conference on Systems Engineering, pp. 348–354, 2005.

[309] S. C. Stubberud, K. A. Kramer, Incorporation of uncertainty into level 2 fusion association metrics, in: Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pp. 205–210, 2005.

[310] P. Svenson, Equivalence classes of future paths for sensor allocation and threat analysis, in: Proceedings of SPIE, 5913:363–369, 2005.

[311] P. Svenson, T. Berg, P. Hörling, M. Malm, C. Mårtenson, Using the impact matrix for predictive situational awareness, in: Proceedings of the International Conference on Information Fusion, Paper 1298, 2007.

[312] P. Svenson, C. Mårtenson, SB-Plan: simulation-based support for resource allocation and mission planning, in: Proceedings of the Conference on Civil and Military Readiness, Paper S3.6, 2006.

[313] K. Sycara, R. Glinton, B. Yu, J. Giampapa, S. Owens, M. Lewis, LTC C. Grindle, An integrated approach to high-level information fusion, Information Fusion, 10(1):25–50, 2009.

[314] K. Sycara, M. Lewis, From data to actionable knowledge and decision, in: Proceedings of the International Conference on Information Fusion, 1:577–584, 2002.

[315] The MathWorks, Inc., Fuzzy Logic Toolbox User's Guide Version 2, 2002.

[316] The MathWorks, Inc., Fuzzy Logic Toolbox Version 2.1.2 (Release 13), 2002.

[317] C. Thomas, N. Balakrishnan, Performance enhancement of intrusion detection systems using advances in sensor fusion, in: Proceedings of the International Conference on Information Fusion, Session SS01, 2008.

[318] C. Thomas, N. Balakrishnan, Modified evidence theory for performance enhancement of intrusion detection systems, in: Proceedings of the International Conference on Information Fusion, Session RS07, 2008.

[319] United States Air Forces, Multi-Command Handbook 11-F16, Volume 5, 1996.

[320] R. van der Merwe, A. Doucet, J. F. G. de Freitas, E. Wan, The Unscented Particle Filter, Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, England, 2000; http://www.cs.ubc.ca/~nando/software.html.

[321] E. Waltz, J. Llinas, Multisensor Data Fusion, Artech House, ISBN: 0890062773, 1990.

[322] Y. Wang, H. Yang, X. Wang, R. Zhang, Distributed intrusion detection system based on data fusion method, in: Proceedings of the World Congress on Intelligent Control and Automation, 5:4331–4334, 2004.

[323] R. B. Washburn, M. K. Schneider, J. J. Fox, Stochastic dynamic programming based approaches to sensor resource management, in: Proceedings of the International Conference on Information Fusion, 1:608–615, 2002.

[324] G. A. Watson, W. D. Blair, IMM algorithm for tracking targets that maneuver through coordinated turns, in: Proceedings of SPIE, 1698:236–247, 1992.

[325] A. M. Waxman, D. A. Fay, B. J. Rhodes, T. S. McKenna, R. T. Ivey, N. A. Bomberger, V. K. Bykoski, G. A. Carpenter, Information fusion for image analysis: geospatial foundations for higher-level fusion, in: Proceedings of the International Conference on Information Fusion, 1:562–569, 2002.

[326] M. Wei, G. Chen, E. Blasch, H. Chen, J. B. Cruz, Jr., Game theoretic multiple mobile sensor management under adversarial environments, in: Proceedings of the International Conference on Information Fusion, Session SS04, 2008.

[327] K. White, J. Williams, P. Hoffensetz, Radar sensor management for detection and tracking, in: Proceedings of the International Conference on Information Fusion, Session SS04, 2008.

[328] M. Witkowski, G. White, P. Louvieris, G. Görbil, E. Gelenbe, L. Dodd, High-level information fusion and mission planning in highly anisotropic threat spaces, in: Proceedings of the International Conference on Information Fusion, Session SS13, 2008.

[329] W. R. Wu, Target tracking with glint noise, IEEE Transactions on Aerospace and Electronic Systems, 29(1):174–185, 1993.

[330] W. R. Wu, Maximum likelihood identification of glint noise, IEEE Transactions on Aerospace and Electronic Systems, 32(1):41–51, 1996.

[331] W. R. Wu, D. C. Chang, Maneuvering target tracking with colored noise, IEEE Transactions on Aerospace and Electronic Systems, 32(4):1311–1320, 1996.

[332] Y. Wu, D. Hu, M. Wu, X. Hu, Unscented Kalman filtering for additive noise case: augmented versus nonaugmented, IEEE Signal Processing Letters, 12(5):357–360, 2005.

[333] Y. Wu, D. Hu, M. Wu, X. Hu, Unscented Kalman filtering for additive noise case: augmented vs. non-augmented, in: Proceedings of the American Control Conference, Portland, Oregon, USA, 6:4051–4055, 2005.

[334] N. Xiong, P. Svensson, Multi-sensor management for information fusion: issues and approaches, Information Fusion, 3(2):163–186, 2002.

[335] Y. Xun, M. M. Kokar, K. Baclawski, Control based sensor management for a multiple radar monitoring scenario, Information Fusion, 5(1):49–63, 2004.

[336] T. Yan, B. Wang, Grid architecture model of network centric warfare, Journal of Systems Engineering and Electronics, 17(1):121–125, 2006.

[337] N. Yang, W. Tian, Z. Jin, An interacting multiple model particle filter for manoeuvring target location, Measurement Science and Technology, 17(6):1307–1311, 2006.

[338] S. J. Yang, A. Stotz, J. Holsopple, M. Sudit, M. Kuhl, High level information fusion for tracking and projection of multistage cyber attacks, Information Fusion, 10(1):107–121, 2009.

[339] J. L. Yepes, I. Hwang, M. Rotea, New algorithms for aircraft intent inference and trajectory prediction, Journal of Guidance, Control, and Dynamics, 30(2):370–382, 2007.

[340] J. Yuill, F. Wu, J. Settle, F. Gong, R. Forno, M. Huang, J. Asbery, Intrusion-detection for incident-response, using a military battlefield-intelligence process, Computer Networks, 34(4):671–697, 2000.

[341] M. Zandipour, B. J. Rhodes, N. A. Bomberger, COALESCE: A probabilistic ontology-based scene understanding approach, in: Proceedings of the International Conference on Information Fusion, Session RS08, 2008.

[342] R. Zhan, J. Wan, Passive maneuvering target tracking using 3D constant-turn model, in: Proceedings of the IEEE Conference on Radar, pp. 404–411, 2006.

[343] X. Zhang, M. Zhang, S. Sun, An artificial intelligent approach on longevity modeling, in: Proceedings of the IEEE International Conference on Tools with Artificial Intelligence, pp. 462–465, 2005.

[344] Y. Zhao, C. Haissig, M. J. Hoffman, Analysis of pilot intent parameters in air traffic management, in: Proceedings of the American Control Conference, 3:1789–1792, 1998.

[345] Q. Zhu, S. L. Aldridge, T. N. Resha, Hierarchical Collective Agent Network (HCAN) for efficient fusion and management of multiple networked sensors, Information Fusion, 8(3):266–280, 2007.

[346] J. H. Zwaga, H. Driessen, Tracking performance constrained MFR parameter control: applying constraints on prediction accuracy, in: Proceedings of the International Conference on Information Fusion, 1:546–551, 2005.

Appendix A

Mathematical and Statistical Results

A.1 Central Limit Theorem and Law of Large Numbers

Theorem A.1 *Central Limit Theorem.*

Let X_i, $i = 1, 2, \ldots$, be a sequence of independent random variables. Then the sample mean

$$Y_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

converges to a Gaussian (normal) distribution as $n \to \infty$.

If X_i, $i = 1, 2, \ldots$, are independent and identically distributed (i.i.d.), each with (the same) finite mean $E(X_i) = \mu$ and variance σ^2, then Y_n converges to $\mathcal{N}(\mu, \sigma^2/n)$ as $n \to \infty$. Define the random variable

$$Z_n := \frac{\sum_{i=1}^{n} X_i - n\mu}{\sigma\sqrt{n}} = \frac{(Y_n - \mu)}{(\sigma/\sqrt{n})}.$$

Then Z_n converges to the standard Gaussian distribution $\mathcal{N}(0, 1)$ in distribution:

$$Z_n \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1) \quad \text{as} \quad n \to \infty.$$

Equivalently,

$$\lim_{n\to\infty} F_n(x) = \Phi(x), \quad x \in \mathbb{R},$$

where $F_n(\cdot)$ and $\Phi(\cdot)$ represent the cumulative distribution functions of Z_n and $\mathcal{N}(0, 1)$ respectively.

□

Theorem A.2 *Strong Law of Large Numbers.*

Let X_i, $i = 1, 2, \ldots$, be a sequence of i.i.d. random variables each with (the same) finite mean $E(X_i) = \mu$. For $n \in \mathbb{N}$, let the sample mean be

$$Y_n = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

Then, the sample mean converges almost surely to the mean. Equivalently, the sample mean converges to the mean with probability 1:

$$Y_n \xrightarrow{a.s.} \mu \quad \text{as} \quad n \to \infty \qquad \text{or} \qquad \text{Prob}\left(\lim_{n\to\infty} Y_n = \mu\right) = 1.$$

□

Theorem A.3 *Weak Law of Large Numbers.*

Let X_i, $i = 1, 2, \ldots$, be a sequence of i.i.d. random variables each with (the same) finite mean $E(X_i) = \mu$. For $n \in \mathbb{N}$, let the sample mean be

$$Y_n = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

Then, the sample mean converges in probability towards the mean:

$$Y_n \xrightarrow{P} \mu \quad \text{as} \quad n \to \infty.$$

Equivalently, for any number $\epsilon > 0$,

$$\lim_{n\to\infty} \text{Prob}\left(|Y_n - \mu| < \epsilon\right) = 1 \qquad \text{or} \qquad \lim_{n\to\infty} \text{Prob}\left(|Y_n - \mu| \geq \epsilon\right) = 0.$$

□

A.2 Fuzzy Logic

Generally, vagueness and imprecision exist in data/information concerning real-world problems. Fuzzy logic [152, 315], an extension of Boolean logic, was developed to deal with uncertainties associated with problems from practical applications.

In *classical set theory*, a set has a crisp (sharp and clear) boundary and it completely includes or excludes an arbitrarily given element. On the other hand, in *fuzzy set theory*, boundaries between sets of values need not be distinctly defined. A fuzzy set expresses the degree to which an element belongs to a set, where an element can have gradual transition in status from "belongs to a set" to "does not belong to a set".

Let X be a space of objects and x be an arbitrary element of X. For a *classical set* C, $C \subseteq X$, define a *characteristic function* $f : X \mapsto \{0, 1\}$ by

$$f(x) = \begin{cases} 0, & x \notin C, \\ 1, & x \in C. \end{cases}$$

Then C can be represented by a set of ordered pairs,

$$C' = \{(x, f(x)) \mid x \in X\}. \tag{A.1}$$

Definition A.1 *Fuzzy sets and membership functions.*

Let X be a space of objects which are generically denoted by x. A *fuzzy set* F in X is defined as a set of ordered pairs

$$F = \{(x, \mu_F(x)) \mid x \in X\}, \tag{A.2}$$

where $\mu_F : X \mapsto Y$ is known as the *membership function* for F. The membership function maps each element x of the *input space* (or *universe of discourse*) X to a *degree of membership* (also known as *membership value* or *membership grade*) $\mu_F(x)$ in the *output space* (or *membership space*) Y. For each $x \in X$, $\mu_F(x) \in [0, 1]$. $\square$

Remark: The definition of a fuzzy set is an extension of the definition of a classical set. In Definition A.1, if $Y = \{0, 1\}$, then F is reduced to a classical set and $\mu_F(\cdot)$ is the characteristic function of F.

Fuzzy logic is a superset of standard Boolean logic. There exist fuzzy logical operations for fuzzy sets that correspond to Boolean logical operations for classical sets. In the case when membership function values are restricted to the set $\{0, 1\}$, fuzzy logical operations and Boolean logical operations are equivalent.

Definition A.2 *Fuzzy complement.*

A *fuzzy complement* operator is a continuous function $N : [0,1] \to [0,1]$ that meets the basic axiomatic requirements:

$$\begin{aligned} &N(0) = 1 \text{ and } N(1) = 0 && \text{(boundary)}, \\ &N(a) \geq N(b) \text{ if } a \leq b && \text{(monotonicity)}. \end{aligned} \tag{A.3}$$

An optional requirement imposes *involution* on a fuzzy complement:

$$N(N(a)) = a \quad \text{(involution)}, \tag{A.4}$$

which guarantees that the double complement of a fuzzy set is still the set itself.

The *complement* of a fuzzy set F is the fuzzy set $\bar{F}$ (or $\neg F$, NOT F), whose membership function is related to that of F by

$$\mu_{\bar{F}}(x) = N(\mu_F(x)), \tag{A.5}$$

with the fuzzy complement operator commonly defined by $N(a) = 1 - a$.

□

Definition A.3 *T-norm.*

A *T-norm* operator is a binary function $T : [0,1] \times [0,1] \to [0,1]$ that satisfies:

$$\begin{aligned} &T(0,0) = 0,\ \ T(a,1) = T(1,a) = a && \text{(boundary)}, \\ &T(a,b) \leq T(c,d) \text{ if } a \leq c \text{ and } b \leq d && \text{(monotonicity)}, \\ &T(a,b) = T(b,a) && \text{(commutativity)}, \\ &T(a, T(b,c)) = T(T(a,b), c) && \text{(associativity)}. \end{aligned} \tag{A.6}$$

□

Definition A.4 *T-conorm (or S-norm).*

A *T-conorm* (or *S-norm*) operator is a binary function $S : [0,1] \times [0,1] \to [0,1]$ satisfying:

$$\begin{aligned} &S(1,1) = 1,\ \ S(0,a) = S(a,0) = a && \text{(boundary)}, \\ &S(a,b) \leq S(c,d) \text{ if } a \leq c \text{ and } b \leq d && \text{(monotonicity)}, \\ &S(a,b) = S(b,a) && \text{(commutativity)}, \\ &S(a, S(b,c)) = S(S(a,b), c) && \text{(associativity)}. \end{aligned} \tag{A.7}$$

□

Definition A.5 *Fuzzy intersection (conjunction).*

The *intersection* of two fuzzy sets F_1 and F_2 is a fuzzy set F, written as $F = F_1 \cap F_2$ or $F = F_1$ AND F_2. F is specified in general by a T-norm operator $T : [0,1] \times [0,1] \to [0,1]$, which aggregates the membership values of F_1 and F_2 as

$$\mu_F(x) = T(\mu_{F_1}(x), \mu_{F_2}(x)). \tag{A.8}$$

A frequently used T-norm operator is defined by $T(a,b) = \min(a,b)$, the *minimum* of $\{a,b\}$ (also denoted by $a \wedge b$).

$\square$

Definition A.6 *Fuzzy union (disjunction).*

The *union* of two fuzzy sets F_1 and F_2 is a fuzzy set F, written as $F = F_1 \cup F_2$ or $F = F_1$ OR F_2. F is specified in general by a T-conorm (or S-norm) operator $S : [0,1] \times [0,1] \to [0,1]$, which aggregates the membership values of F_1 and F_2 as

$$\mu_F(x) = S(\mu_{F_1}(x), \mu_{F_2}(x)). \tag{A.9}$$

A frequently used S-norm operator is defined by $S(a,b) = \max(a,b)$, the *maximum* of $\{a,b\}$ (also denoted by $a \vee b$).

$\square$

For an input vector $x \in X$, a fuzzy inference process utilizes a set of fuzzy rules to interpret the values of x and assign appropriate values to an output vector $y \in Y$. Each rule is of the form "if S_1 then S_2", or equivalently, "$S_1 \to S_2$". The if-part of the rule "S_1" is called the *antecedent*, while the then-part of the rule "S_2" is called the *consequent*. Each rule outputs a fuzzy set. Aggregation of the output fuzzy sets for the rules yields a single output fuzzy set. Defuzzification is carried out on the resultant set to obtain the final desired conclusion, in the form of a single number.

A.3 Derivation of Equations 3.67 and 3.68

The Jacobian H of the measurement equation in Section 3.5.1 is evaluated at the predicted state at each time step. In the derivation that follows, the time index is omitted.

By Equations 3.65 and 3.66, the (i,j)-entry of H is denoted by

$$H_{ij} = \frac{\partial h[i]}{\partial X[j]} = \frac{\partial Z[i]}{\partial X[j]}, \quad i = 1,2, \quad j = 1,2,$$

where $h[k]$, $X[k]$ and $Z[k]$ are the k-th entry of the vectors $h(X)$, X and Z respectively. Explicitly,

$$\begin{aligned} H_{11} &= \frac{\partial Z[1]}{\partial X[1]} = \frac{\partial c}{\partial \lambda}, & H_{12} &= \frac{\partial Z[1]}{\partial X[2]} = \frac{\partial c}{\partial \sigma}, \\ H_{21} &= \frac{\partial Z[2]}{\partial X[1]} = \frac{\partial p}{\partial \lambda}, & H_{22} &= \frac{\partial Z[2]}{\partial X[2]} = \frac{\partial p}{\partial \sigma}. \end{aligned} \tag{A.10}$$

It is noted that the standard Gaussian pdf $N'(\cdot)$ is an even function, that is, for any real number x,

$$N'(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} = \frac{1}{\sqrt{2\pi}} e^{-(-x)^2/2} = N'(-x). \tag{A.11}$$

By Equation 3.63,

$$\frac{\partial d_1}{\partial \lambda} = \frac{\partial d_2}{\partial \lambda} \qquad \text{and} \qquad \frac{\partial d_1}{\partial \sigma} = \frac{\partial d_2}{\partial \sigma} + \sqrt{\psi}. \tag{A.12}$$

By Equations 3.61 and A.12,

$$\begin{aligned} \frac{\partial c}{\partial \lambda} &= S \frac{\partial N(d_1)}{\partial \lambda} - X \frac{\partial (e^{-\lambda\psi} N(d_2))}{\partial \lambda} \\ &= S N'(d_1) \frac{\partial d_1}{\partial \lambda} - X \left[-\psi e^{-\lambda\psi} N(d_2) + e^{-\lambda\psi} \frac{\partial N(d_2)}{\partial \lambda} \right] \\ &= S N'(d_1) \frac{\partial d_1}{\partial \lambda} + X\psi e^{-\lambda\psi} N(d_2) - X e^{-\lambda\psi} N'(d_2) \frac{\partial d_2}{\partial \lambda} \\ &= X\psi e^{-\lambda\psi} N(d_2) + \left[S N'(d_1) - X e^{-\lambda\psi} N'(d_2) \right] \frac{\partial d_1}{\partial \lambda}. \end{aligned} \tag{A.13}$$

By Equations 3.62, A.11 and A.12,

$$\begin{aligned} \frac{\partial p}{\partial \lambda} &= -S \frac{\partial N(-d_1)}{\partial \lambda} + X \frac{\partial (e^{-\lambda\psi} N(-d_2))}{\partial \lambda} \\ &= -S N'(-d_1) \frac{\partial (-d_1)}{\partial \lambda} + X \left[-\psi e^{-\lambda\psi} N(-d_2) + e^{-\lambda\psi} \frac{\partial N(-d_2)}{\partial \lambda} \right] \\ &= -S N'(d_1) \left(-\frac{\partial d_1}{\partial \lambda} \right) - X\psi e^{-\lambda\psi} N(-d_2) + X e^{-\lambda\psi} N'(-d_2) \frac{\partial (-d_2)}{\partial \lambda} \\ &= S N'(d_1) \frac{\partial d_1}{\partial \lambda} - X\psi e^{-\lambda\psi} N(-d_2) + X e^{-\lambda\psi} N'(d_2) \left(-\frac{\partial d_2}{\partial \lambda} \right) \\ &= -X\psi e^{-\lambda\psi} N(-d_2) + \left[S N'(d_1) - X e^{-\lambda\psi} N'(d_2) \right] \frac{\partial d_1}{\partial \lambda}. \end{aligned} \tag{A.14}$$

By Equations 3.61 and A.12,

$$\begin{aligned}\frac{\partial c}{\partial \sigma} &= S\frac{\partial N(d_1)}{\partial \sigma} - Xe^{-\lambda\psi}\frac{\partial N(d_2)}{\partial \sigma} \\ &= SN'(d_1)\frac{\partial d_1}{\partial \sigma} - Xe^{-\lambda\psi}N'(d_2)\frac{\partial d_2}{\partial \sigma} \\ &= SN'(d_1)\left(\frac{\partial d_2}{\partial \sigma} + \sqrt{\psi}\right) - Xe^{-\lambda\psi}N'(d_2)\frac{\partial d_2}{\partial \sigma} \\ &= S\sqrt{\psi}N'(d_1) + \left[SN'(d_1) - Xe^{-\lambda\psi}N'(d_2)\right]\frac{\partial d_2}{\partial \sigma}.\end{aligned} \tag{A.15}$$

By Equations 3.62, A.11 and A.15,

$$\begin{aligned}\frac{\partial p}{\partial \sigma} &= -S\frac{\partial N(-d_1)}{\partial \sigma} + Xe^{-\lambda\psi}\frac{\partial N(-d_2)}{\partial \sigma} \\ &= -SN'(-d_1)\frac{\partial(-d_1)}{\partial \sigma} + Xe^{-\lambda\psi}N'(-d_2)\frac{\partial(-d_2)}{\partial \sigma} \\ &= -SN'(d_1)\left(-\frac{\partial d_1}{\partial \sigma}\right) + Xe^{-\lambda\psi}N'(d_2)\left(-\frac{\partial d_2}{\partial \sigma}\right) \\ &= SN'(d_1)\frac{\partial d_1}{\partial \sigma} - Xe^{-\lambda\psi}N'(d_2)\frac{\partial d_2}{\partial \sigma} \\ &= \frac{\partial c}{\partial \sigma}.\end{aligned} \tag{A.16}$$

To prove: $SN'(d_1) = Xe^{-\lambda\psi}N'(d_2)$.

By Equation 3.63,

$$\begin{aligned}d_1^2/2 &= d_2^2/2 + d_2\sigma\sqrt{\psi} + \sigma^2\psi/2 \\ &= d_2^2/2 + [\ln(S/X) + (\lambda - \sigma^2/2)\psi] + \sigma^2\psi/2 \\ &= d_2^2/2 + \ln(S/X) + \lambda\psi.\end{aligned} \tag{A.17}$$

By Equations A.11 and A.17,

$$\begin{aligned}SN'(d_1) = S\frac{1}{\sqrt{2\pi}}e^{-d_1^2/2} &= S\frac{1}{\sqrt{2\pi}}e^{-d_2^2/2-\ln(S/X)-\lambda\psi} \\ &= S\frac{1}{\sqrt{2\pi}}e^{-d_2^2/2}e^{-\ln(S/X)}e^{-\lambda\psi} \\ &= SN'(d_2)(X/S)e^{-\lambda\psi} \\ &= Xe^{-\lambda\psi}N'(d_2).\end{aligned} \tag{A.18}$$

■

From the result in Equation A.18, together with Equations A.10, A.13, A.14, A.15 and A.16, one gets the required expressions for the matrix entries of

$$H = \begin{bmatrix} \frac{\partial c}{\partial \lambda} & \frac{\partial c}{\partial \sigma} \\ \frac{\partial p}{\partial \lambda} & \frac{\partial p}{\partial \sigma} \end{bmatrix},$$

Specifically,

$$\frac{\partial c}{\partial \lambda} = X\psi e^{-\lambda\psi}N(d_2), \qquad \frac{\partial p}{\partial \lambda} = -X\psi e^{-\lambda\psi}N(-d_2), \qquad \frac{\partial c}{\partial \sigma} = \frac{\partial p}{\partial \sigma} = S\sqrt{\psi}N'(d_1).$$

□

Appendix B

List of Publications

Part of the work done in this thesis have been reported in the following papers:

1. G. W. Ng, K. H. Ng, R. Yang, and P. H. Foo, *Intent inference for attack aircraft through fusion*, in: Proceedings of SPIE, Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2006, Orlando, Florida, USA, 19-20 April 2006, Volume 6242, 624206.

2. P. H. Foo, and G. W. Ng, *Combining IMM method with particle filters for 3D maneuvering target tracking*, in: Proceedings of the 10th International Conference on Information Fusion, Québec, Canada, 9-12 July 2007, Paper 1131.

3. P. H. Foo, G. W. Ng, K. H. Ng, and R. Yang, *Application of intent inference for surveillance and conformance monitoring to aid human cognition*, in: Proceedings of the 10th International Conference on Information Fusion, Québec, Canada, 9-12 July 2007, Paper 1309.

4. P. H. Foo, G. W. Ng, K. H. Ng, and R. Yang, *Application of intent inference for air defense and conformance monitoring*, Journal of Advances in Information Fusion, accepted for publication, 2009.

Work on the following paper is ongoing:

P. H. Foo, and G. W. Ng, *Combining the interacting multiple model method with particle filters for three-dimensional manœuvring target tracking*, in progress.